An overview of targeted attacks and APTs on Linux

By GReAT

Published: 2020-09-10 · Archived: 2026-04-05 19:59:04 UTC

Perhaps unsurprisingly, a lot has been written about targeted attacks on Windows systems. Windows is, due to its popularity, the platform for which we discover most APT attack tools. At the same time, there's a widely held opinion that Linux is a secure-by-default operating system that isn't susceptible to malicious code. It's certainly true that Linux hasn't faced the deluge of viruses, worms and Trojans faced by those running Windows systems over the years. However, there is certainly malware for Linux – including PHP backdoors, rootkits and exploit code. Moreover, numbers can be misleading. The strategic importance of servers running Linux makes them an attractive target for attackers of all kinds. If an attacker is able to compromise a server running Linux, they not only gain access to data stored on the server but can also target endpoints connected to it running Windows or macOS – for example, through a drive-by download. Furthermore, Linux computers are more likely to be left unprotected, so that such a compromise might well go unnoticed. When the Heartbleed and Shellshock vulnerabilities were first reported in 2014, two major concerns were that compromised Linux servers could become an attacker's gateway into a corporate network and could give an attacker access to sensitive corporate data.

The Global Research and Analysis Team (GReAT) at Kaspersky publishes regular summaries of advanced persistent threat (APT) activity, based on the threat intelligence research discussed in greater detail in our private APT reports. In this report, we focus on the targeting of Linux resources by APT threat actors.

Readers who would like to learn more about our intelligence reports or request more information on a specific report are encouraged to contact intelreports@kaspersky.com.

Barium

We first wrote about the Winnti APT group (aka APT41 or Barium) in 2013, when they were targeting mostly gaming companies for direct financial profit. Meanwhile, they grew their operations, developed tons of new tools and went for much more complex targets. MESSAGETAP is Linux malware used by this group to selectively intercept SMS messages from the infrastructure of telecoms operators. According to FireEye, the group deployed this malware on SMS gateway systems as part of its operations to infiltrate ISPs and telecoms companies in order to build a surveillance grid.

Recently, we discovered another suspected Barium/APT41 tool, written in the programming language Go (also known as Golang) that implements a dynamic, C2-controlled packet corruption/network attack tool for Linux machines. Although it's not 100% clear if this is a tool developed for system administration tasks or if it is also part of the APT41 toolset, the fact that the functionality it offers can also be achieved through other system management tools suggests that its purpose may not be legitimate. Also, its name on disk is rather generic and is unrelated to its functionality, again suggesting that it is potentially a covert tool used for carrying out certain types

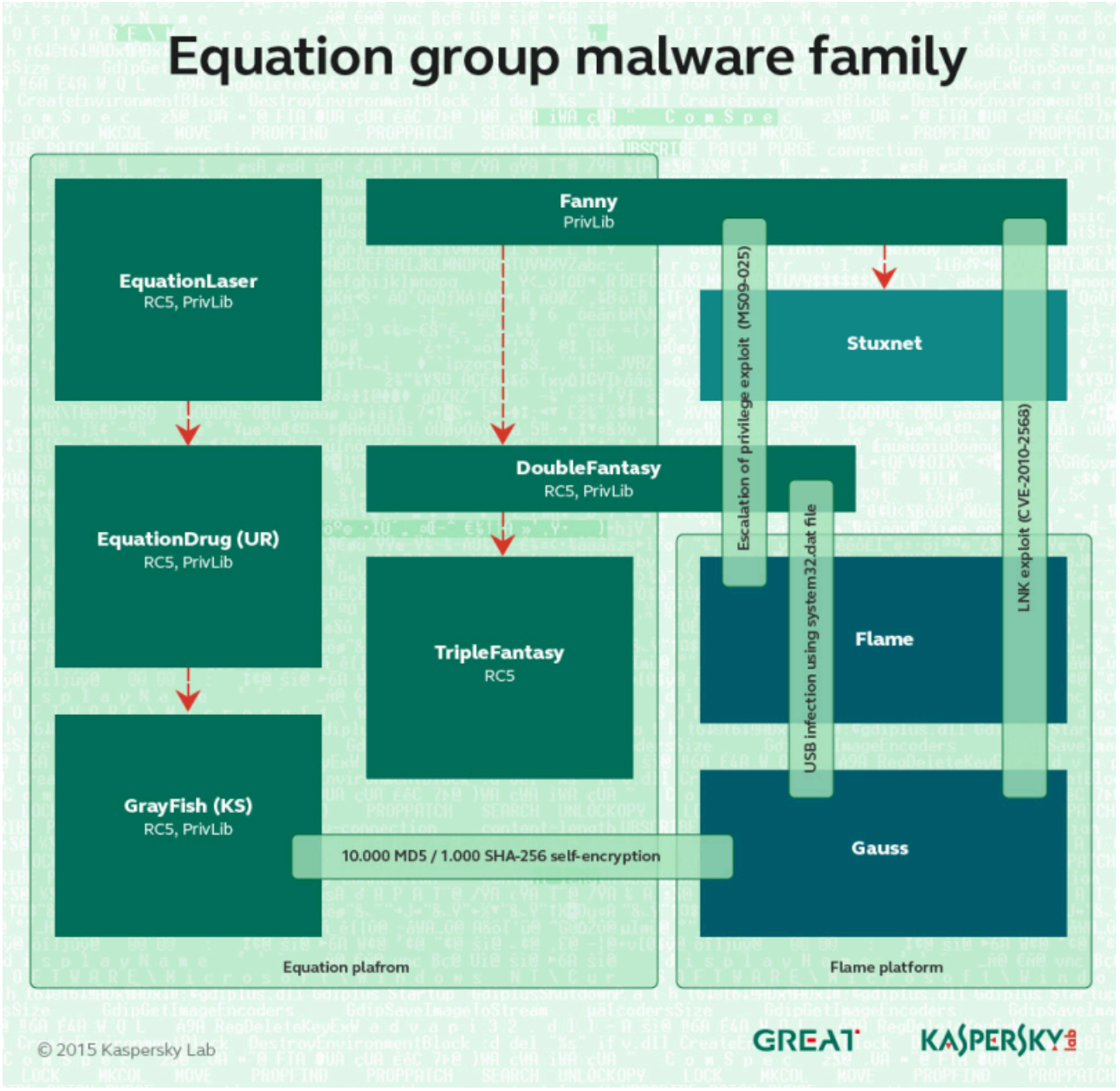
of destructive attacks. More details about this tool can be found in our private report “Suspected Barium network control tool in GO for Linux”.

Cloud Snooper

In February 2020, Sophos published a report describing a set of malicious tools it attributes to a previously unknown threat actor called Cloud Snooper. The centerpiece is a server-oriented Linux kernel rootkit that hooks netfilter traffic control functions in order to enable firewall-traversing covert C2 (command-and-control) communications. We analyzed and described the rootkit’s userland companion backdoor, dubbed ‘Snoopy’, and were able to design detection and scanning methods to identify the rootkit at scale. We also discovered more samples, as well as targeted servers in Asia. We believe that this evolved toolset might have been in development since at least 2016.

Equation

We uncovered the Equation group in 2015. This is a highly sophisticated threat actor that has been engaged in multiple CNE (computer network exploitation) operations dating back to 2001, and perhaps as early as 1996. For many years this threat actor interacted or worked together with other powerful APT groups, for projects such as Stuxnet and Flame. The group has a powerful arsenal of implants. Among those we found were: ‘EQUATIONLASER’, ‘EQUATIONDRUG’, ‘DOUBLEFANTASY’, ‘TRIPLEFANTASY’, ‘FANNY’ and ‘GRAYFISH’. The innovations of the Equation group aren’t limited to the Windows platform. The group’s POSIX-compliant codebase allows for parallel developments on other platforms. In 2015, we came by the early-stage DOUBLEFANTASY malware for Linux. This implant collects system information and credentials and provides generic access to an infected computer. Given the role this module plays in the infection lifecycle, it would suggest the presence of analogous later-stage, more sophisticated implants, although we weren’t able to find any.



HackingTeam

HackingTeam was an Italian information technology company that developed and sold intrusion and so called "legal surveillance software" to governments, law enforcement agencies and businesses around the world. Unfortunately for them, they were hacked and suffered a data breach in 2015, at the hands of the activist known as Phineas Phisher. The subsequent leak of 400GB of stolen company data, including source code and customer information, allowed these tools to be acquired, adapted and used by threat actors around the world, such as DancingSalome (aka Callisto). The leaked tools included a zero-day exploit for Adobe Flash (CVE-2015-5119) as well as sophisticated platforms capable of providing remote access, keylogging, general information recording and exfiltration, and perhaps most notably, the ability to retrieve Skype audio and video frames directly from memory, bypassing stream encryption. The RCS (Remote Control System) malware (aka Galileo, Da Vinci, Korablin, Morcut and Crisis) includes multiple components, including desktop agents for Windows, macOS and perhaps unsurprisingly… Linux.

Lazarus

In late 2018, we discovered a previously unknown malicious framework that we named MATA internally. This framework was used to target commercial companies in Korea, India, Germany and Poland. While we weren't able to find code overlaps with any other known actor, the Kaspersky Threat Attribution engine showed code similarities with Manuscrypt, complex malware used by Lazarus (aka Hidden Cobra). This framework, as with earlier malware developed by Lazarus, included a Windows backdoor. However, we also found a Linux variant that we believe was designed for networking devices.

In June 2020, we analyzed new macOS samples linked to Lazarus Operation AppleJeus and TangoDaiwbo campaigns, used in financial and espionage attacks. The samples had been uploaded to VirusTotal. The uploaded files also included a Linux malware variant that included similar functionality to the macOS TangoDaiwbo malware. These samples confirm a development that we had highlighted two years earlier – that the group was actively developing non-Windows malware.

Sofacy

Sofacy (aka APT28, Fancy Bear, STRONTIUM, Sednit and Tsar Team) is a highly active and prolific APT threat actor. From its high-volume zero-day deployment to its innovative, broad malware set, Sofacy is one of the top groups that we monitor. Among the tools in the group's arsenal is SPLM (also known as CHOPSTICK and XAgent), a second-stage tool used selectively against targets around the world. Over the years, Sofacy has developed modules for several platforms, including, in 2016, modules for Linux, detected as 'Fysbis'. The consistent artefacts seen over the years and across Windows, macOS, iOS and Linux suggests that the same developers, or a small core team, is modifying and maintaining the code.

The Dukes

The Dukes is a sophisticated threat actor that was first documented by us in 2013, but whose tools have been used in attacks dating back to 2008. The group is responsible for attacks against targets in Chechnya, Ukraine, Georgia, as well as western governments and NGOs, NATO and individuals – the group is thought to be behind the hack of the Democratic National Congress in 2016. The Dukes' toolset includes a comprehensive set of malware implementing similar functionality but coded in several different programming languages. The group's malware and campaigns include PinchDuke, GeminiDuke, CosmicDuke, MiniDuke, CozyDuke, OnionDuke, SeaDuke, HammerDuke and CloudDuke. At least one of these, SeaDuke, includes a Linux variant.

The Lamberts

The Lamberts is a highly sophisticated threat actor group which is known to possess a huge malware arsenal, including passive, network-driven backdoors, several generations of modular backdoors, harvesting tools and wipers for carrying out destructive attacks. We created a color scheme to distinguish the various tools and implants used against different victims around the world.

Lamberts discovery timeline

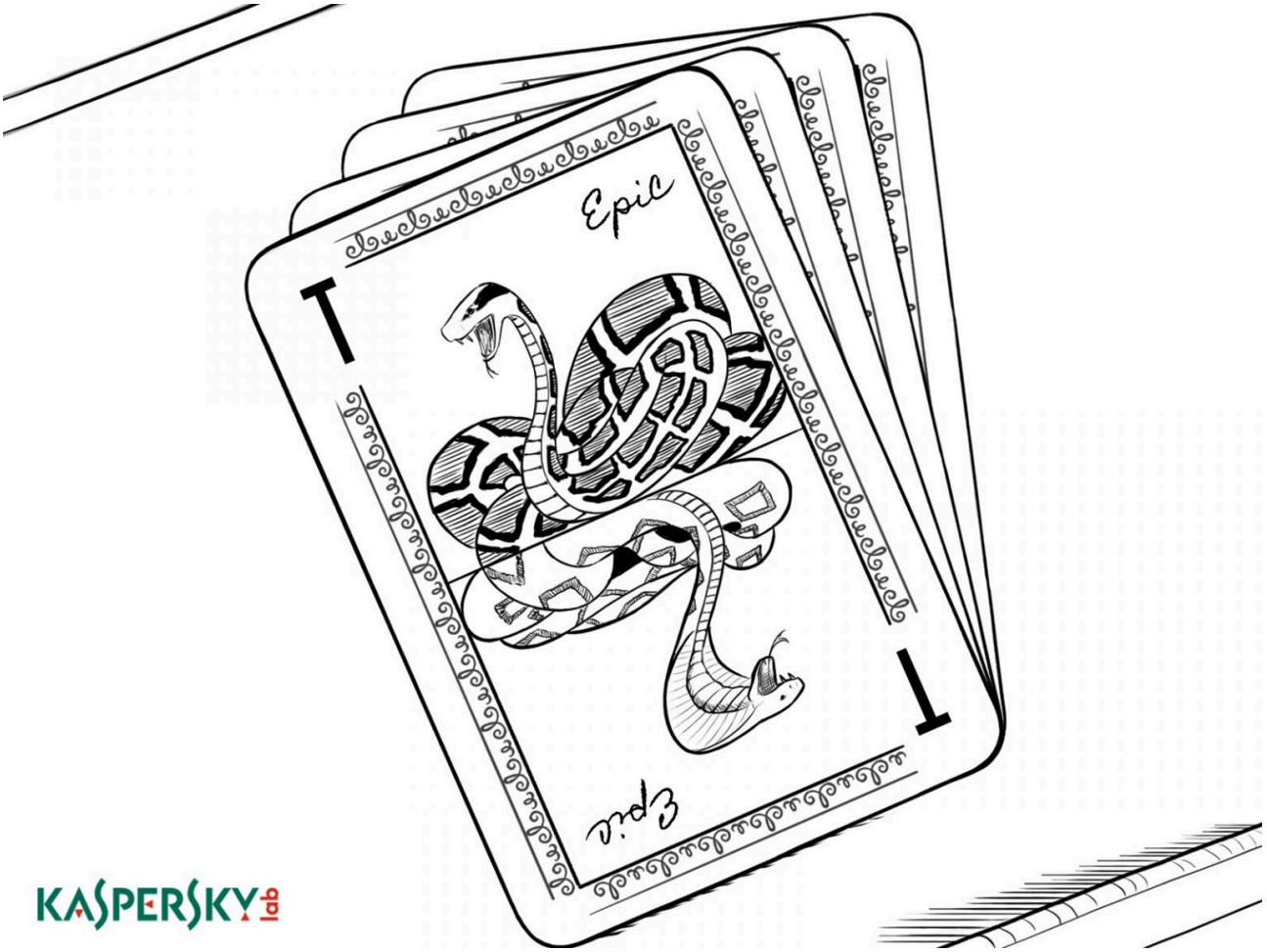
In 2017, we published an overview of the Lamberts family; and further updates (GoldLambert, SilverLambert, RedLambert, BrownLambert) are available to customers of our threat intelligence reports. The focus of the various Lamberts variants is definitely Windows. Nevertheless, signatures that we created for Green Lambert for Windows also triggered on a macOS variant of Green Lambert that was functionally similar to the Windows version. In addition, we also identified samples of the SilverLambert backdoor compiled for both Windows and Linux.

Tsunami backdoor

Tsunami (aka Kaiten) is a UNIX backdoor used by multiple threat actors since it was first seen in the wild in 2002. The source code was made public some years ago; and there are now more than 70 variants. The source code compiles smoothly on a wide range of embedded devices; and there are versions for ARM, MIPS, Sparc and Cisco 4500/PowerPC. Tsunami remains a threat for Linux-based routers, DVRs and the increasing number of IoT (internet of things) devices. In 2016, a variant of Tsunami was used in the Linux Mint hack, where an unknown threat actor compromised the Linux Mint distribution ISOs to include a backdoor. We also observed the use of the Tsunami backdoor to surgically target a number of cryptocurrency users on Linux.

Turla

Turla (aka Uroboros, Venomous Bear and Waterbug) is a prolific Russian-speaking group known for its covert exfiltration tactics such as the use of hijacked satellite connections, water-holing of government websites, covert channel backdoors, rootkits and deception tactics. This threat actor, like other APT groups, has made significant changes to its toolset over the years. Until 2014, every malware sample used by Turla that we had seen was designed for 32- or 64-bit versions of Windows.



Then in December 2014, we published our report on Penguin Turla, a Linux component in the Turla arsenal. This is a stealth backdoor that didn't require elevated privileges, i.e. administrator or root rights. Even if someone with limited access to the system launches it, the backdoor can intercept incoming packets and run commands from the attackers on the system while maintaining stealth. It is also rather hard to uncover, so if it's installed on a compromised server, it could sit there unnoticed for a long time. Further research on Penguin Turla revealed that its roots stretch back to the Moonlight Maze operation in the mid-1990s. In May this year, researchers from Leonardo published a report about Penguin_x64, a previously undocumented variant of the Penguin Turla Linux backdoor. Based on this report, we generated network probes that detect Penquin_x64 infected hosts at scale, allowing us to discover a couple dozen infected servers in Europe and the US, as recent as July 2020. We believe that, following public documentation of GNU/Linux tools, Turla may have been repurposing Penguin to conduct operations other than traditional intelligence gathering.

Two-Sail Junk

In January 2020, a watering hole was discovered that utilized a full remote iOS exploit chain to deploy a feature-rich implant named LightSpy. The site appears to have been designed to target users in Hong Kong, based on the content of the landing page. For the time being, until we can link the campaign to a known group, we have given the name Two-Sail Junk to the threat actor behind this implant. However, while our public report focused on the iOS implant, the project is broader than previously thought, supporting an Android implant, and probably supporting implants for Windows, Linux and MacOS.

WellMess

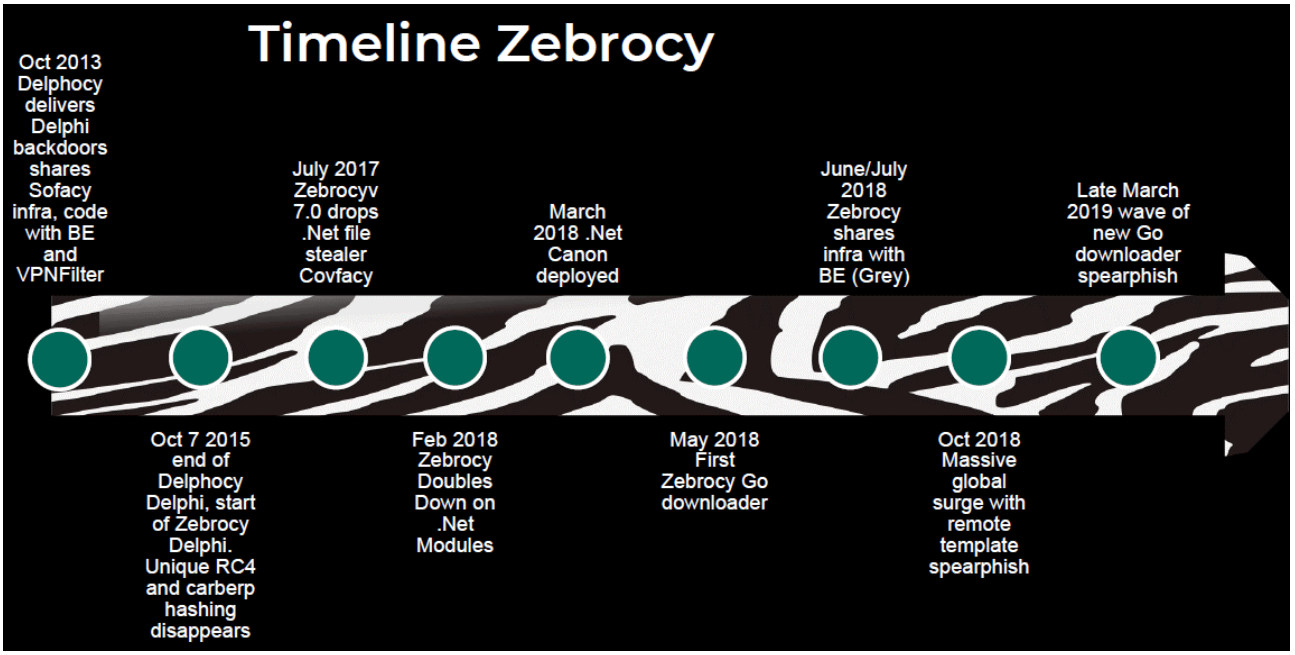
In March 2020, we began to actively track new C2 servers associated with malware commonly referred to as WellMess, indicating a potentially massive new wave of activity. This malware was initially documented by JPCERT in July 2018 and has been sporadically active since then. There were rumors that hint at a possible connection with CozyDuke (aka APT29), along with speculation that the current activity was focused on the healthcare industry, although we were unable to verify either claim. WellMess is a Remote Access Trojan, written in .NET and Go (Golang), cross-compiled to be compatible with both Windows and Linux.

WildNeutron

We first published about WildNeutron in 2015, together with our colleagues from Symantec, who call it Morpho or Butterfly. This group, which rose to prominence with their 2012-2013 attacks on Twitter, Microsoft, Apple and Facebook, are one of the most elusive, mysterious and dynamic we have seen. Their arsenal included many interesting and innovative tools, such as LSA backdoors or IIS plugins, coupled with both zero-day-based and physical deployment. Unsurprisingly, in several known attacks WildNeutron used a custom Linux backdoor as well.

Zebrocy

Zebrocy is custom malware that we have been tracking since 2015. The group using this malware started as a subset of Sofacy, but also has similarities and overlaps with other APT groups. The group has developed malware in several languages, including Delphi, AutoIT, .NET, C#, PowerShell and Go. Zebrocy has mainly targeted Central Asian government-related organizations, both in-country and in remote locations. The group makes extensive use of spear phishing to compromise Windows endpoints. However, its backdoors are configured to communicate directly with IP-assigned web server hosts over port 80; and the group seems to favor Linux for this part of its infrastructure – specifically, Apache 2.4.10 running on Debian Linux.



Recommendations for protecting Linux systems

One of the main reasons that Linux systems go unprotected is a false sense of security from using Linux instead of the far more popular (and more targeted) Windows. Nevertheless, we hope all the aforementioned points are convincing enough for you to start securing your Linux-based machines in a serious way.

The very first recommendation is to **maintain a list of trusted sources** of your software. Think about this in the same way as the recommended approach to Android or iOS apps – only installing applications from official repositories. In the Linux world we enjoy more freedom: for example, even if you are using Ubuntu, you're not restricted only to Canonical's own repository. Any .DEB file, or even application source code from GitHub, is at your service. But please choose these sources wisely. Don't just blindly follow instructions like "Run this script from our server to install"; or "curl https://install-url | sudo bash" – which is a security nightmare.

Please also be mindful of the **secure way to get applications** from these trusted repositories. Your channels to update the apps have to be encrypted using HTTPS or SSH protocols. Besides your trust in software sources and its delivery channel, it's critical for **updates to arrive in a timely fashion**. Most modern Linux flavors are able to do this for you, but a simple cron script would help you to stay more protected and to get all the patches as soon as they are released by developers.

The next thing we would recommend is checking network-related settings. With commands like "netstat -a" you could **filter out all unnecessary opened ports** on your host. Please avoid network applications you really don't need or don't use to minimize your network footprint. Also, it would be strongly recommended to **properly set up the firewall** from your Linux distributive, to filter traffic and store the host's network activity. It's also a very good idea not to go online directly, but through NAT.

To continue with the network-related security rules, we recommend **protecting your locally stored SSH keys** (used for your network services) using passwords at least. In more "paranoid" mode you could even **store the keys on external protected storage**, like tokens from any trusted vendor. On the server side of connections, nowadays it's not that hard to **set up multi-factor authentication for SSH sessions**, like the messages to your phone or other mechanisms such as authenticator apps.

So far, our recommendations have covered software sources, application delivery channel, avoiding unnecessary network footprint and protection of encryption keys. One more idea we recommend for monitoring threats you couldn't find at the filesystem level is to **keep and analyze the network activity logs**. You could install and use an out-of-band network tap to independently monitor and analyze the network communications of your Linux systems.

As part of your threat model, you need to consider the possibility that, despite all the aforementioned measures, attackers can compromise your protection. Think about the next protection step in terms of an attacker's persistence in the system. They will probably make changes to be able to start their Trojan automatically after the system reboots. So, you need to **regularly monitor the main configuration files as well as the integrity of system binaries**, just in case of file viruses. The logs mentioned above for monitoring network communication, is fully applicable here: **the Linux auditing system collects system calls and file access records**. Additional daemons such as "osquery" can be used for the same task. . Any suspicious files, URLs, and IP addresses can be checked at Kaspersky Threat Intelligence Portal.

Physical security of devices is also important. It doesn't matter how much attention you pay to network and system level hardening if your laptop ends up in an attacker's hands and you haven't taken steps to protect it from this attack vector. You should consider **full disk encryption and safe boot mechanisms** for physical security. A more spy-like approach would be to place tamper-evident security tape on your most critical hardware.

Dedicated solution with Linux security can simplify the protection task: web threat protection detects malicious and phishing websites; network threat protection detects network attacks in incoming traffic; behavior analysis detects malicious activity, while device control allows management of connected devices and access to them.

Our final recommendation relates to Docker. This is not a theoretical threat: infection of containers is a very real issue. **Containerization doesn't provide security by itself**. Some containers are quite isolated from the host, but not all – **network and file system interfaces exist** in them and in most cases there are bridges between physical and containerized worlds.

Therefore, you can use security solution that allows to add security into development process. Kaspersky Hybrid Cloud Security includes integration with CI/CD platforms, such as Jenkins, through a script to scan Docker images for malicious elements at different stages.

To prevent supply-chain attacks, On-Access Scanning (OAS) and On-Demand Scanning (ODS) of containers, images, and local and remote repositories can be used. Namespace monitoring, flexible mask-based scan scope control and the ability to scan different layers of containers help to enforce secure development best practices.

We have broken down this list of recommendations into logical sections. Please bear in mind that, besides applying all the measures we have mentioned, you should also audit and check all the generated logs and any other messages regularly. Otherwise you could miss signs of intrusion. A final idea, for security enthusiasts, is to adopt active measures – to provide system penetration testing from time to time.

Summary of recommendations:

- Maintain a list of trusted software sources, avoid using unencrypted update channels.
- Do not run binaries and scripts from untrusted sources. A widely advertised way to install programs with commands like "curl https://install-url | sudo bash" is a security nightmare.
- Make sure your update procedure is effective. Set up automatic security updates.
- Spend time to set up your firewall properly: make sure it logs network activity, block all ports you don't use, minimize your network footprint.
- Use key-based SSH authentication, protect keys with passwords.
- Use 2FA and store sensitive keys on external token devices (e.g. Yubikey).
- Use an out-of-band network tap to independently monitor and analyze network communications of your Linux systems.
- Maintain system executable file integrity. Review configuration file changes regularly.
- Be prepared for insider/physical attacks: use full disk encryption, trusted/safe boot and put tamper-evident security tape on your critical hardware.
- Audit the system, check logs for indicators of attacks.
- Run penetration tests on your Linux setup.

- Use a dedicated security solution for Linux with web and network protection, as well as features for DevOps protection.

Source: https://securelist.com/an-overview-of-targeted-attacks-and-apts-on-linux/98440/