ESET Research White Papers // April 2018

Protecting Enterprise

An examination of bugs, major vulnerabilities and exploits

Author

Tony Anscombe

Contributing researchers

Anton Cherepanov
Aryeh Goretsky
Ondrej Kubovič
Robert Lipovský
Miguel Ángel Mendoza
Diego Perez

ENJOY SAFER TECHNOLOGY™

CONTENTS

EXECUTIVE SUMMARY

This white paper focuses on the dramatic growth in the number and severity of software vulnerabilities, and discusses how multilayered endpoint security is needed to mitigate the threats they pose.

Exploits of critical vulnerabilities such as EternalBlue have been utilized to devastating effect. In 2017, EternalBlue alone spawned WannaCryptor, CoinMiner and Diskcoder (aka Petya). In 2018 the security community has come to realize the extent of CPU architecture vulnerabilities. Also there is a growing acceptance that most older infrastructure is "vulnerable by design". Furthermore, exploits frequently take advantage of known vulnerabilities that have already been patched, but whose updates have not been installed across whole organizations. Both WannaCryptor and Diskcoder affected organizations worldwide despite operating system updates being available. ESET detected and blocked malware taking advantage of the EternalBlue exploit.

The purpose of this white paper is to help users understand why no single technology or mix of technologies will guarantee that a network will not be compromised and why the cybersecurity industry, including ESET, constantly refines products both reactively and proactively, adding layers to ensure effective security.

BUGS, VULNERABILITIES & EXPLOITS

It is almost a truism that all complex software has bugs, but does all software with bugs have vulnerabilities?

A software bug is an error in the design, implementation or execution of a computer program. These errors cause the program to behave in unexpected ways, such as producing incorrect output or behavior. There are formal methods that can prove the design and implementation of at least some kinds of software perfectly meet their specifications (that is, the resulting programs will do everything that they should, and nothing that they should not). However, such approaches tend to be prohibitively expensive for all but the simplest software projects. Further, it is unlikely that much of the codebase that our existing systems are built from either will be refactored through such methodologies any time soon, or even can be re-engineered practically with these methods.

Thus, it seems we are likely to be saddled with software bugs for some time to come. And that means we will also be saddled with the attendant costs. Indeed, a report published by the US National Institute of Standards and Technology (NIST) in 2002 estimated the likely cost of software bugs to the US economy that year would be *$59.5 billion* – about 0.6% of the country's GDP!

Of course, times have changed and our lives, both personal and business, have become increasingly digitalized since 2002 – for example, the first iPhone was not released until 29 June, 2007. As we show in the next section, both the number of software vulnerabilities and their severity have increased over time, so it is likely these costs will be markedly higher than that now.

So, most software contains bugs. A program that, among other functions, calculates 2 + 2 = 5 is certainly buggy, but is it vulnerable? Quite possibly not... In computer security, software vulnerabilities are usually described in terms of computer program flaws that could allow an adversary to compromise the confidentiality, integrity, or availability of a program or its data, or that of the broader computer system on which it runs. Programs that only exhibit bugs such as simple mathematical calculation errors, as in the hypothetical example above, may not have any security-relevant code or system exposure, so those bugs would not expose any security vulnerabilities.

Odds are that this document is being read on a device such as a phone, tablet or laptop, using a PDF viewer app or program. That viewer is software, so as already discussed, it probably has bugs, but what about security vulnerabilities? Does it have any, and more importantly, is it safe to use?

Addressing the "is the program vulnerable?" issue first, much recent history suggests that this viewer program will have some security vulnerabilities, but how can we find out? That is a conundrum – for most of their lives, software bugs, whether they expose security vulnerabilities or not, are latent "features" of a program. They lie in the code, undetected and unknown, until some odd, unexpected program behavior is noticed. Or, perhaps a security researcher systematically probes all possible code branches in some of the program's functions, or disassembles the program's code, and notices some misfeature or "undocumented functionality".

Regardless of how, once a bug is uncovered it may be investigated further, to decide if it exposes a security vulnerability. If it does, and this is reported to the program's developer, hopefully an update that fixes the bug and removes the vulnerability is developed promptly and offered to the program's users. During that process, however, users of the program normally remain oblivious to the vulnerability's existence. Users usually only learn of vulnerabilities once the program's developer publicly announces that security updates are available. Such disclosures commonly occur in a security advisory, announcing the update's availability and providing some information about the vulnerabilities it addresses. In that period between a vulnerability's discovery and the public availability of a security update, it may be referred to as a "zero-day vulnerability".

An important distinction to understand is the difference between a vulnerability and an exploit for that vulnerability. Security researchers who discover software vulnerabilities often develop exploits for them. An exploit is typically program code, or a set of procedures, that demonstrates not only that a bug exists, but also that it is a security vulnerability because it can be used to compromise the confidentiality, integrity, or availability of the affected program or its data. Sometimes exploits are publicly released, often after the vulnerability is disclosed, ostensibly to allow other security researchers to test the vulnerability's existence and/or to confirm that mitigations and updates actually work. Sometimes an exploit is publicly disclosed before the vendor has released updates that fix the vulnerability. Such exploits may be referred to as "zero-day exploits".

Addressing the second of those questions, above – "is it safe to use?" – raises the thorny issue of security risk assessment. A whole book could be written on this topic, and indeed, many have been. We will leave risk assessment as an exercise for the reader, for now, and continue with our focus on vulnerabilities, exploits and how endpoint security products can help mitigate the threats they pose.

THE VULNERABILITY TREND

Vulnerabilities are one of the elements frequently identified in security incidents together with other threats like *exploits* and *malware*. In 2017, not only did the number of security vulnerabilities reported reach a historic peak but the number identified as "high" or "critical" severity also reached a peak.

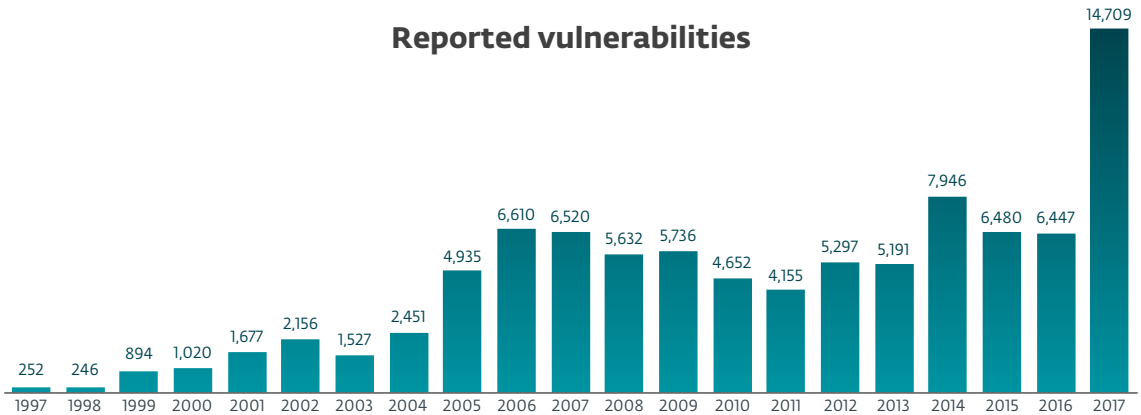
According to *CVE Details*, an independent source that tracks vulnerabilities, more than 14,700 vulnerabilities were reported in 2017, compared to 6,447 in 2016 *(see chart below)*.

That is an increase of almost 130% over 2016, and an average of 40 vulnerabilities per day, compared to just 17 per day in 2016.

This number does not include vulnerabilities that did not receive an official CVE number. An explosion in the number of security vulnerabilities being disclosed caught the official CVE register off guard in 2016, which may, at least partly, account for the significant increase in 2017. *A report by Risk Based Security* claims that approximately 7,900 reported vulnerabilities remained without a CVE number, leaving them outside the visibility of many IT security departments, but readily located by a motivated adversary.

The increase over the past years could be due in part to the growth in the use of open-source code, both in application software and in devices in the Internet of Things (IoT) category. Open-source code is used in more than 90% of all software, it exists in operating systems, productivity software, administration tools, development tools and code libraries, and third-party developers often either use it to build, or as part of, their solutions. This is compounded by the growth in IoT devices, which commonly use embedded Linux and adjacent open-source components as their operating systems or within their application software, or in firmware on the device.

Another factor to consider is the severity of these vulnerabilities. This is usually determined on the basis of various factors, such as their impact on the confidentiality, integrity or availability of data, as well as which attack vector is used, the complexity of effecting an attack, the privileges required, and whether any user interaction is required. Several systems have been proposed for calculating the overall value of these effects.



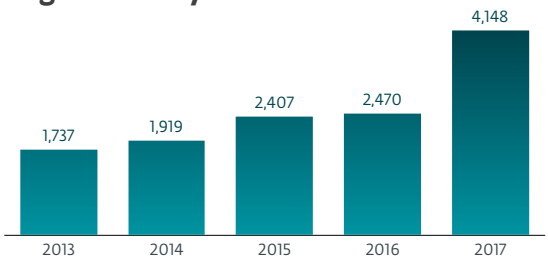
Vulnerabilities that have been assigned a CVE will also have a Common Vulnerability Score System (CVSS) severity score. This is a scoring system designed to provide an open, standardized method for assessing the severity of vulnerabilities. Currently, two versions of this system are in use: CVSS v2.0 and CVSS v3.0.

CVSS incorporates three groups of metrics used to calculate the overall score in a given location. The first group, called the base group, represents the intrinsic qualities of the vulnerability, in other words, those which are inherent to it and do not change over time or across environments. The second group, known as the temporal group, reflects the characteristics that change over time. And lastly, the group of environmental metrics takes into account the characteristics of a vulnerability that are unique to the context of the user carrying out the assessment.

After assigning values to the base metrics, the formula results in a score between 0.0 and 10.0, which represents the base severity score of the vulnerability in question. CVSS scores are split into ranges with qualitative descriptors such as "low", "medium" and "high" attached.

The level of growth for vulnerabilities rated as "high" by their CVSS v2.0 score has been considerable. For example, they increased from 2,470 in 2016 to 4,148 during 2017, representing an increase of 68%.

High severity vulnerabilities



In the data shown above we can see a major increase in the number of vulnerabilities reported in recent years, together with an additional increase in those considered as high severity.

MAJOR SECURITY VULNERABILITIES & ATTACKS

Over the past year there have been a number of major incidents that have either directly threatened and attacked networks and endpoints, or disclosed a significant risk that requires mitigation.

The previous section described the increase, over the previous year, in the vulnerabilities disclosed in 2017. It also revealed the significant number of disclosed vulnerabilities that were not assigned a CVE number, and thus are unlikely to receive much attention in corporate IT security departments. With such growth in the number of disclosed vulnerabilities, it should not be a surprising consequence that cybercriminals are looking to exploit them for their own gain.

Understanding the methods cyberattackers use is important. In this section we detail some examples of the security threats, and attacks, that took place in 2017 and early 2018 that exploited major vulnerabilities.

EternalBlue

EternalBlue is the name of the exploit that enabled WannaCryptor's ability to self-replicate and rapidly spread across networks. Reputedly, the US National Security Agency (NSA) developed this exploit.

A cache of NSA cyberweapons was allegedly stolen by a hacking group known as Shadow Brokers. The group tried to auction these cyberweapons but as the endeavor did not prove profitable they decided to sell the NSA tools individually.

On 14 March, 2017, Microsoft released MS17-010, fixing critical SMB vulnerabilities. At the time, it was not evident that the patch was in any way related to NSA cyberweapons. It was not until 14 April, 2017 – the day on which the Shadow Brokers publicly released many of the stolen tools – that it became clear.

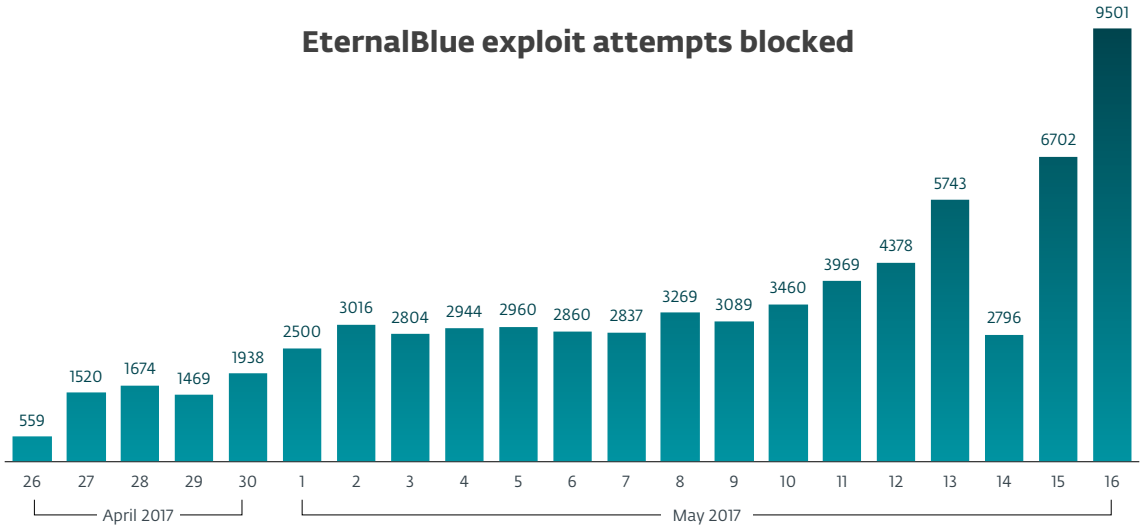
The timing between the patching and the public release of the exploits raised some speculation about the circumstances of the events.

Nevertheless, the situation was as follows: the vulnerability exploited by EternalBlue was patched and updates for it made available via Windows Update a month before knowledge of the exploit itself was made public.

With the exploit publicly released, the race between patching still-not-updated systems and exploiting them started. It was not by chance that the EternalBlue exploit was increasingly used *(see chart below)*.

On 12 May, 2017, EternalBlue became an important component of the already-mentioned massive infection incident. All the elements were in the cybercriminals' hands: the ransomware (WannaCryptor) and the exploit (EternalBlue). There's a period missing after the closing parenthesis. This is laid out in the timeline, below.

Companies could have circumvented the success of this ransomworm in many ways. Firstly, two months beforehand, the patch for EternalBlue was made available. Although it is not related to the ransomware's encryption



routines, it blocks the host from damage caused by another compromised host on the same network. Furthermore, proactive protection installed on the host, such as an exploit blocker or an up-to-date endpoint security product could block infection attempts, and stop the attack in cases where the malware breaks through the network's defenses.

News about a "kill switch" feature in the ransomware came out. It was noticed, by an independent researcher, that the malware made an HTTP request that was supposed to fail before moving onto the encryption routine. Since the domain was previously unregistered, all requests were failing, consequently enabling the ransomware to encrypt files. After registering the requested domain the researchers were able to redirect requests to replying servers and stop the (first variant of the) ransomworm's propagation.

In the UK many National Health Service sites suffered due to the malware exploiting a vulnerability in older Windows operating systems where no update was available. The update was available for Microsoft customers with these operating systems and customers who were paying for extended support since March 2017. On May 12, at the peak of the outbreak, *Microsoft decided to make the same update available* to all affected users, including those with out-of-support OS.

It did not take long for other variants to show up. New versions appeared that patched the previous domain and versions without the kill switch were released. It was made evident once more that exploiting vulnerabilities (not necessarily zero-days) can have a huge impact on normal business operation.

These fast responses by cybercriminals show that combating malware is no easy task. They strike quickly and are highly adaptive.

ESET detected and blocked attempts to exploit this Windows vulnerability as far back as 25 April, 2017 – and thus protected its customers before the malicious payload, WannaCryptor, was even created.

WannaCryptor

WannaCryptor, also referred to as WannaCry or Wcrypt, spread rapidly. One reason for the speed at which the malware spread is the way it utilized the EternalBlue exploit.

Unlike most crypto-ransomware, WannaCryptor has wormlike capabilities, allowing it to spread by itself. As a result, it spread very quickly *(see image below)*.

The story started in Spain's telecom sector and quickly spread from that point, onward and outward. Reports of healthcare related organizations being affected in the UK began to appear, quickly followed by reports of various commercial websites, entire enterprise sites, and just about every type of network in between. People from around the world posted screenshots of the malware from computers in offices, hospitals, and schools.

The worst issue dealt with by victims was that files touched by the attack were encrypted with the attacker being the only source for the key needed to reverse that. This had dire consequences, especially in the healthcare sector. Encrypted patient records, doctor's files and other items may not have been usable or accessible *without* a good backup to restore from.

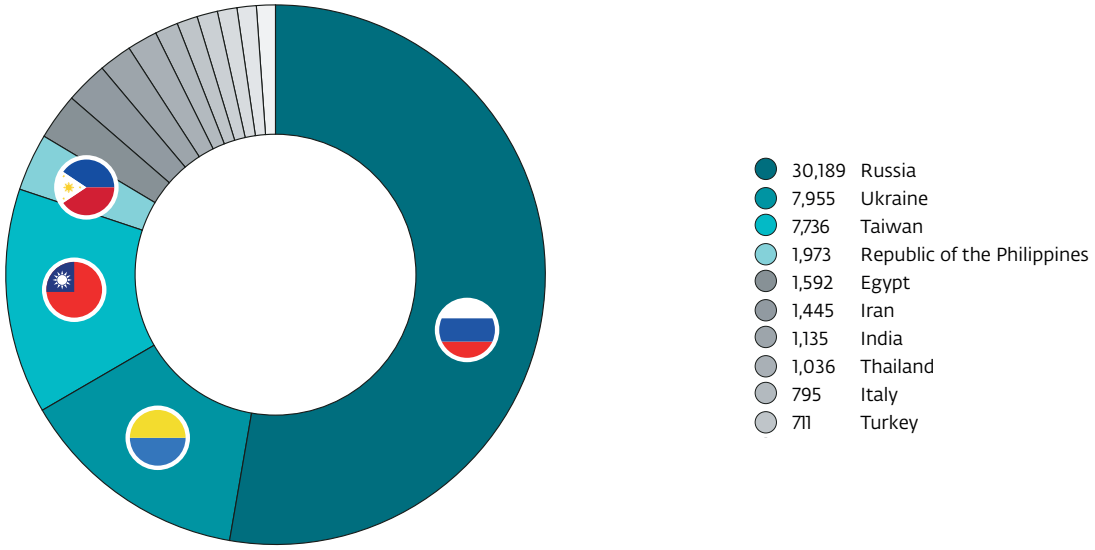
The ransom demanded for decryption of the files was $300 in bitcoin, which is lower than other ransomware we have seen, but the true cost was the time, lost files, and other collateral damage caused by this malware *(see chart below)*.

The chaos that ensued after the WannaCryptor global outbreak seemed to motivate other black hats to scale up their efforts too. We witnessed a significant increase in the number of malicious emails sent out by other notorious actors such as Nemucod, spreading other ransomware such as Filecoder.FV.

WannaCryptor used the EternalBlue exploit to take advantage of the Windows vulnerability. As previously mentioned, ESET detected and blocked attempts to exploit this as far back as 25 April, 2017. Once blocked from exploiting a vulnerability it is important to remember that it is possible cybercriminals will look for alternative delivery methods, thus detection should be integrated in all security layers that have the potential to stop an attack, regardless of the delivery method.

WannaCryptor detections were added to ESET products on 6 April, 2017, when the malware family was first seen. It was not until 12 May, 2017 that it would become a household name when, combined with the EternalBlue exploit, it become a major global infection.

This is an excellent example of a layered security approach, especially when one of the layers is focused on the underlying vulnerability itself. Independent testing organization, MRG Effitas, conducted a *test* the day after the WannaCryptor outbreak and found that by that point in time only three of the mainstream security products they tested blocked the EternalBlue exploit; ESET was one of them.



CoinMiner

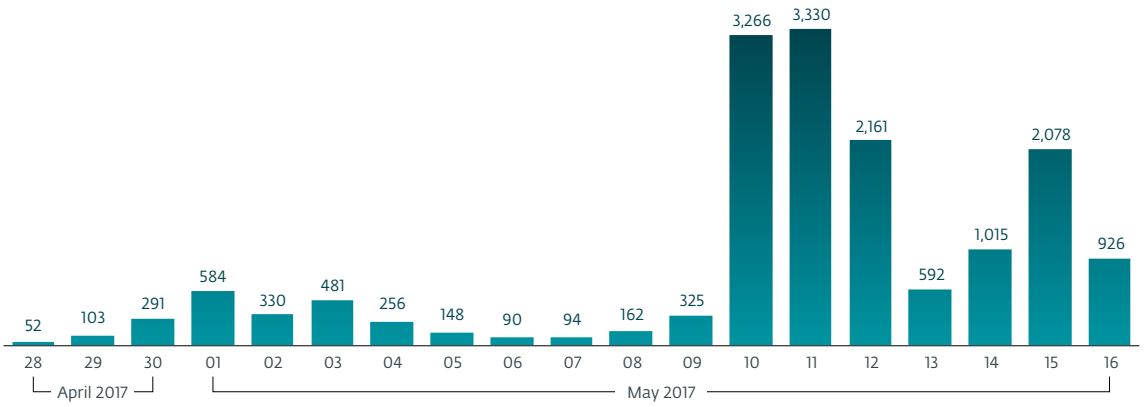
WannaCryptor was not the only malware event misusing the EternalBlue exploit, leaked by Shadow Brokers.

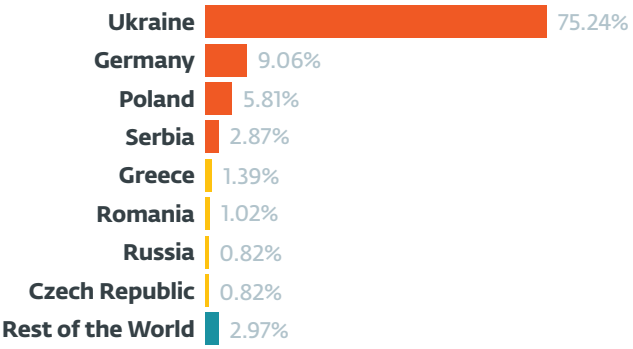
The same mechanisms had been misused by black hats as early as the end of April, when they opted for off-the-shelf Monero cryptocurrency mining software instead of a file-encryption payload. This campaign, detected as Win32/CoinMiner.AFR and Win32/CoinMiner.AFU, started only a few days after the NSA tools leaked online.

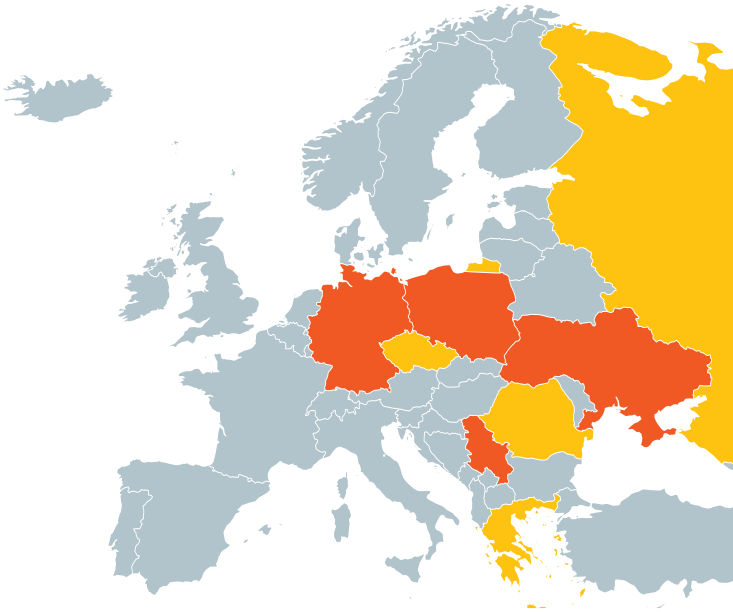
We saw the biggest uptick in EternalBlue detections within roughly 48 hours before the worldwide WannaCryptor ransomware outbreak. Mining malware detections increased from hundreds to thousands of detections per day. We witnessed attempts in as many as 118 countries, with Russia, Taiwan and Ukraine topping the list.

The mining software consumed system resources so intensively that in some cases it rendered the compromised machines unresponsive. This crypto-mining attack blocked port 445, used by the EternalBlue exploit, closing the door for any future compromise using the same vector – including WannaCryptor. If the miners had not taken this precaution, the number of WannaCryptor infections could have been even greater than reported.

EternalBlue+CoinMiner.AFR/.AFU detections







Diskcoder (aka Petya)

In July 2017, there was another global cyberattack, detected by ESET security products as Win32/Diskcoder.C, and variously called Petya, NotPetya, NyetPetya, etc by others. As with WannaCryptor, it highlighted the reality that outdated systems and insufficient security solutions are still widespread.

Diskcoder has a similar impact to WannaCryptor, preventing access to information stored in a system. However, it not only encrypted the information on vulnerable computers, but, following system restart, it left the operating system unusable, encrypting the master boot record so that victims are forced to perform a reinstallation.

Both Diskcoder and WannaCryptor use the leaked NSA exploit called EternalBlue. However this is where the similarity ends. Diskcoder implements other propagation techniques by abusing legitimate Microsoft Windows tools, such as PsExec, and Windows Management Instrumentation Command-line (WMIC), a utility for managing data and functionality on local and remote computers running Windows operating systems.

After the malware is run, it creates a scheduled task to restart the computer within a certain timeframe, which is usually no more than 60 minutes. In addition, it verifies whether or not there are shared folders or disks to which the malware can propagate. If there are, it uses WMIC to run the malicious software on remote machines.

It then starts encrypting files with certain extensions. We should highlight that, unlike most ransomware, it does not change or add a particular extension after encrypting each file, which is a technique widely used by attackers to distinguish encrypted files. In addition, this malware tries to delete event logs to leave no trace, as well as hide its actions.

In the following screenshot you can see the file extensions that the malware will attempt to encrypt:

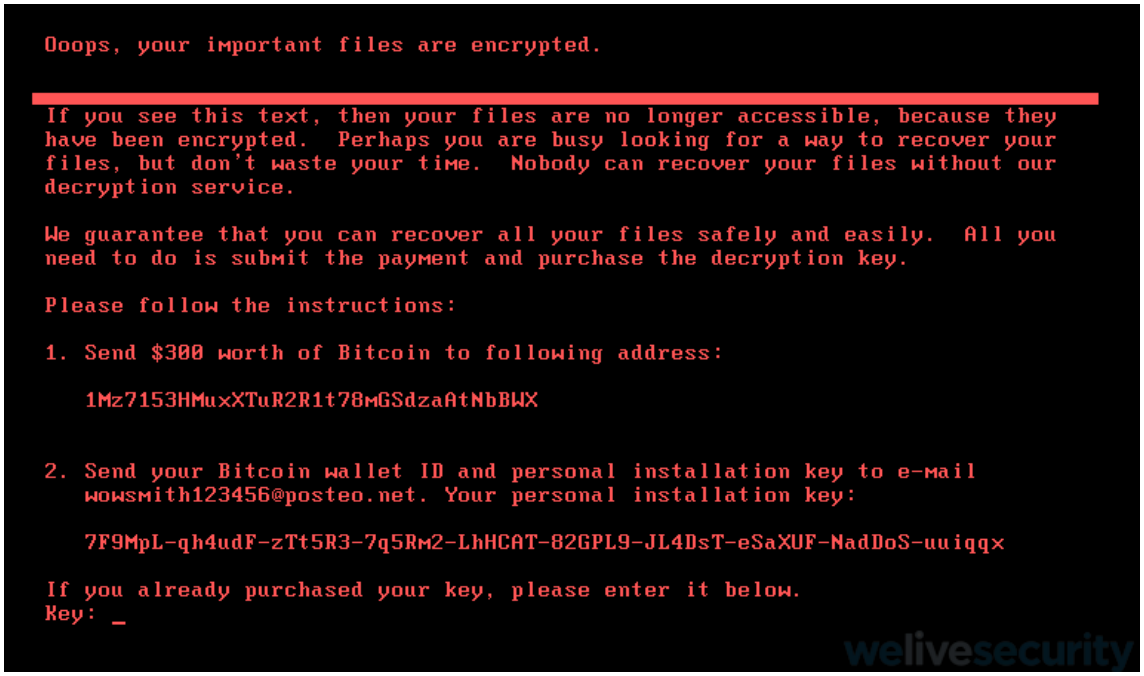
```
.rdata:10010840 a_3ds_7z_accdb_:                        ; DATA XREF: .text:10001B0A↑o
.rdata:10010840                                         ; .data:10018BD4↓o
.rdata:10010840                 unicode 0, <.3ds.7z.accdb.ai.asp.aspx.avhd.back.bak.c.cfg.conf.cpp.cs>
.rdata:10010840                 unicode 0, <.ctl.dbf.disk.djvu.doc.docx.dwg.eml.fdb.gz.h.hdd.kdbx.mai>
.rdata:10010840                 unicode 0, <l.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.pmf.ppt.pptx.pst.pv>
.rdata:10010840                 unicode 0, <i.py.pyc.rar.rtf.sln.sql.tar.vbox.vbs.vcb.vdi.vfd.vmc.vmd>
.rdata:10010840                 unicode 0, <k.vmsd.vmx.vsdx.vsv.work.xls.xlsx.xvd.zip.>,0
.rdata:10010A5E                 align 4
.rdata:10010A60 ; const WCHAR aWs_
```

As previously mentioned, propagation is a prominent characteristic of this threat. Once it manages to infect a computer it attempts to extract the user's credentials and then use them with PsExec and WMIC to search for shared folders and disks. It then spreads via the computer network. In this way, it manages to infect computers located in different countries and regions.

Once the encryption is completed, a set of instructions is displayed, via which the attackers request a payment of $300 worth of bitcoin.

There are no command and control servers connected to the threat in order to trace the perpetrators, or for the peretrators to actually provide decryption should the ransom be paid.

In addition, the currency for the ransom payment is bitcoin, which is practically impossible to trace to its final destination.



Meltdown & Spectre

The first few days of 2018 were filled with anxious discussions concerning widespread and wide-ranging CPU architecture vulnerabilities. Although most of the early concern focused on processors based on Intel's Core architecture, used in PCs for many years, and processors from AMD, the scope of this class of vulnerabilities is much more wide-ranging. Many of the ARM processors, particularly high-end models commonly used in tablets and smartphones, IBM POWER processors used in supercomputers, and even SPARC v9 processors from the mid-1990s, are also affected.

The issue is that programs running in user-mode address space (the "normal" range of memory in which application software, games and the like run) can infer or "see" some of the information stored in kernel-mode address space (the "protected" range of memory that should isolate typical users from operating system code, its device drivers, and sensitive information such as passwords and cryptography certificates).

Vendors quickly released fixes to prevent user-mode programs from "peering inside" kernel-mode memory, unfortunately the initial fixes had the side effect of slowing down operating systems. While the exact amount of slowdown is open to debate Intel stated the performance penalty would "not be significant" for most users.

Processor manufacturer AMD announced that it was unaffected, according to a *message to the Linux Kernel Mailing List* by an AMD engineer, but reports from both Google's Project Zero and Microsoft stated that AMD processors were affected. Both AMD and NVIDIA announced that their GPUs are not vulnerable, although the latter has issued software updates to its device drivers for operating systems affected by the vulnerabilities.

This was not a Windows-specific issue, and it affected Android, Chrome OS, iOS and macOS, among other OSes, and demonstrates that no platform escapes potential vulnerabilities. And the differing communications showed that vendors may not always fully understand if they are vulnerable.

Updating operating systems and processor microcode is a complex process. On 9 January, 2018, Microsoft suspended the Windows update for some older AMD CPUs due to compatibility issues. This was followed on 13 January, 2018, with *Dell*, *Lenovo* and *VMware* suspending some of their microcode updates due to reports of issues after installation.

The confusion over brands of affected CPUs may be due to the fact that this is not one vulnerability, but two similar vulnerabilities, dubbed *Meltdown* and *Spectre* by their respective discoverers. The Meltdown vulnerability is limited to Intel's processors, while Spectre affects AMD, ARM, IBM, Intel and other processors as well.

For many years, processor manufacturers – such as Intel – have been able to fix flaws in processor architecture through microcode updates, which write an update to the processor itself to fix a bug.

Intel's security advisory lists 44 affected families of processors, each of which can contain dozens of models. ARM Limited has released an advisory titled *Vulnerability of Speculative Processors to Cache Timing Side-Channel Mechanism* that lists ten affected models of processor.

In January and into early February both Microsoft and Intel released security updates to mitigate the two vulnerabilities.

THE RISK TO INFRASTRUCTURE

Stuxnet, discovered back in 2005, is malware that caused substantial damage to Iran's nuclear program by interfering with the operation of its Uranium enrichment centrifuges. The malware targeted programmable logic controllers (PLCs) that are used to control machinery, in this case the machinery used for separating nuclear material. While no author or state has ever claimed responsibility it is widely believed to have been state sponsored. The malware took advantage of several zero-day vulnerabilities in software installed on numerous components to take control of the PLCs.

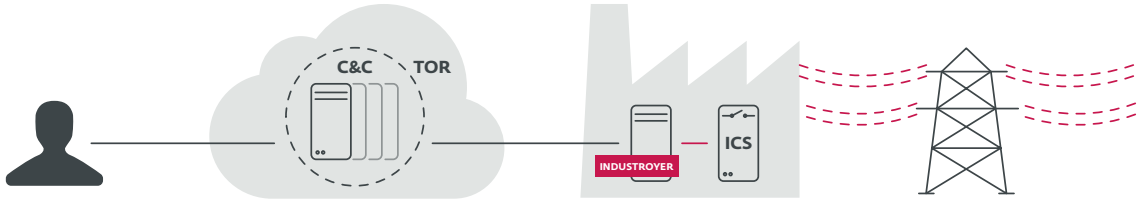
It is important to understand that attacks can happen when vulnerabilities are not present, or maybe more correctly, when the correct protections or security-by-design have not been built into a system from the outset. When cybercriminals find a method to infect a non-vulnerable system it is probably reasonable to say the system was vulnerable by design. Attacking PLCs or industrial control units associated with infrastructure could sound like the world of espionage and possibly too much of a reach for a paper aimed at the commercial sector, but it is important that we appreciate that the buildings which companies rely on have infrastructure (for example, air conditioning systems) and that the protection of the working environment we operate in is possibly as important as protecting the data and systems that run the business.

The following example, documented by ESET's research team in 2017 demonstrates the devastating effect such an attack can produce.

The 2016 attack on Ukraine's power grid that deprived part of its capital, Kiev, of power for an hour was caused by a cyberattack. ESET researchers analyzed samples of malware, detected by ESET as Win32/Industroyer, capable of performing exactly that type of attack. The malware is capable of doing significant harm to electric power systems and could also be refitted to target other types of critical infrastructure *(see infographic below)*.

Industroyer is a particularly dangerous threat, since it is capable of controlling electricity substation switches and circuit breakers directly. To do so, it uses industrial communication protocols used worldwide in power supply infrastructure, transportation control systems, and other critical infrastructure systems (such as water and gas).

These switches and circuit breakers are the digital equivalents of analogue switches; technically they can be engineered to perform various functions. Thus, the potential impact may range from simply turning off power



distribution, cascading failures and more serious damage to equipment. The severity may also vary from one substation to another, as well. Needless to say, disruption of such systems can directly or indirectly affect the functioning of vital services.

Industroyer's gravity lies in the fact that it uses protocols in the way they were designed to be used. The problem is that these protocols were designed decades ago, and back then industrial systems were meant to be isolated from the outside world. Thus, their communication protocols were not designed with security in mind. That means that the attackers did not need to look for protocol vulnerabilities; all they needed was to teach the malware to "speak" those protocols.

The power outage occurred on 17 December, 2016, almost exactly one year after the well-documented cyberattack that caused a blackout that affected around 250,000 households in several regions in Ukraine on 23 December, 2015.

Industroyer is highly customizable malware. While being universal, in that it can be used to attack any industrial control system using some of the targeted communication protocols, several of the components in analyzed samples were designed to target particular hardware. For example, the wiper component and one of the payload components are tailored for use against systems incorporating certain industrial power control products by ABB, and the DoS component works specifically against Siemens SIPROTECT devices used in electrical substations and other related fields of application.

While in principle it is difficult to attribute malware attacks without performing an onsite incident response, it is highly probable that Industroyer was used in the December 2016 attack on the Ukrainian power grid. On top of the fact that the malware clearly possesses the unique capabilities to perform the attack, it contains an activation timestamp for 17 December, 2016, the day of the power outage.

The 2016 attack on the Ukrainian power grid attracted much less attention than the attack that occurred a year earlier. However, the tool most likely used, Win32/Industroyer, is an advanced piece of malware in the hands of a sophisticated and determined attacker.

Thanks to its ability to persist in the system and provide valuable information for tuning-up the highly configurable payloads, attackers could adapt the malware to any environment, which makes it extremely dangerous. Regardless of whether or not the December 2016 attack on the Ukrainian power grid was a test, it should serve as a wake-up call for those responsible for security of critical systems around the world.

PROTECTING THE ENTERPRISE

Earlier in this document we explored some of the most recent major exploits and vulnerabilities; in this section we will explore how different security layers can protect the enterprise.

There is no one technology or mix of technologies that will guarantee that a network will not be compromised. But there are precautions that an organization can take to make sure it has the best possible forms of defense when attacked by a malicious actor.

The concept of a layered security architecture is well established. Layered security, providing "defense in depth", is commonly used in workplaces and home environments to keep assets safe. Offices have doors, doors have locks, then there is the reception desk, building security, building passes, car park passes, and perhaps more distant site perimeter access checks. The point is there is limited access to sensitive parts of the organization, all of which create layers in order to protect the company, its employees and most importantly the customers' data.

When considering cybersecurity, the different layers required to protect the enterprise need to be diverse and are probably provided by multiple vendors, using a best-of-breed approach to selection. Layers might include workstation authentication, content control, policy-based encryption, backup and restore, end point protection, and so on. There are too many layers to mention and in this paper we are looking specifically at vulnerabilities and exploits, and how to protect against them. What is not always apparent though are the layers within a layer, when visualizing endpoint security products there is a common misconception that they work on signatures, whereas in today's sophisticated environment this could not be further from reality.

Most malware is written with the aim of monetization or information theft, and serious money is invested in its development by both criminals and governments. In the hope of making detection more difficult, malware is written in different programming languages, using different compilers and interpreted languages. Code is obfuscated and protected using customized software to make detection and analysis harder. Code is injected into clean processes in an attempt to avoid behavioral monitoring – which is designed to spot suspicious activity – and to hamper removal, thus ensuring persistence in the system. Scripts are used to avoid application control techniques and "in-memory only" or "file-less" malware bypasses file-based security measures.

Malware authors may choose to flood the Internet with thousands of variants of their malware, or alternatively distribute malware to a very limited number of targets, in attempts to avoid attracting the attention of security companies. To avoid detection, clean software components are misused, or malicious code is signed using certificates stolen from legitimate companies. These are just some tactics used to make unauthorized code harder to spot.

At the network level malware makes less use of hardcoded command and control servers to send instructions and receive data from compromised systems. Decentralized control of botnets using peer-to-peer networking is commonly used, and encrypted communication makes identification of attacks harder. Domain generation algorithms reduce the effectiveness of detection based on blocking known domain names. Attackers also take control of legitimate websites that have good reputations, and even legal advertising services are used to serve up malicious content.

In the remainder of this section we will explore some of the technologies used by ESET to mitigate the vulnerabilities and exploits mentioned earlier in this paper.

Updating (aka Patching)

First things first. Updating or patching is similar in concept to fixing a puncture in a bicycle tire. In the context of security, "patches" are issued by companies when security flaws are uncovered.

By way of a more specific definition, a security patch is an update to a piece of software or program to fix a bug or vulnerability, as well as a way to improve it. The same concept as taping up a hole in a tire, but in the digital world.

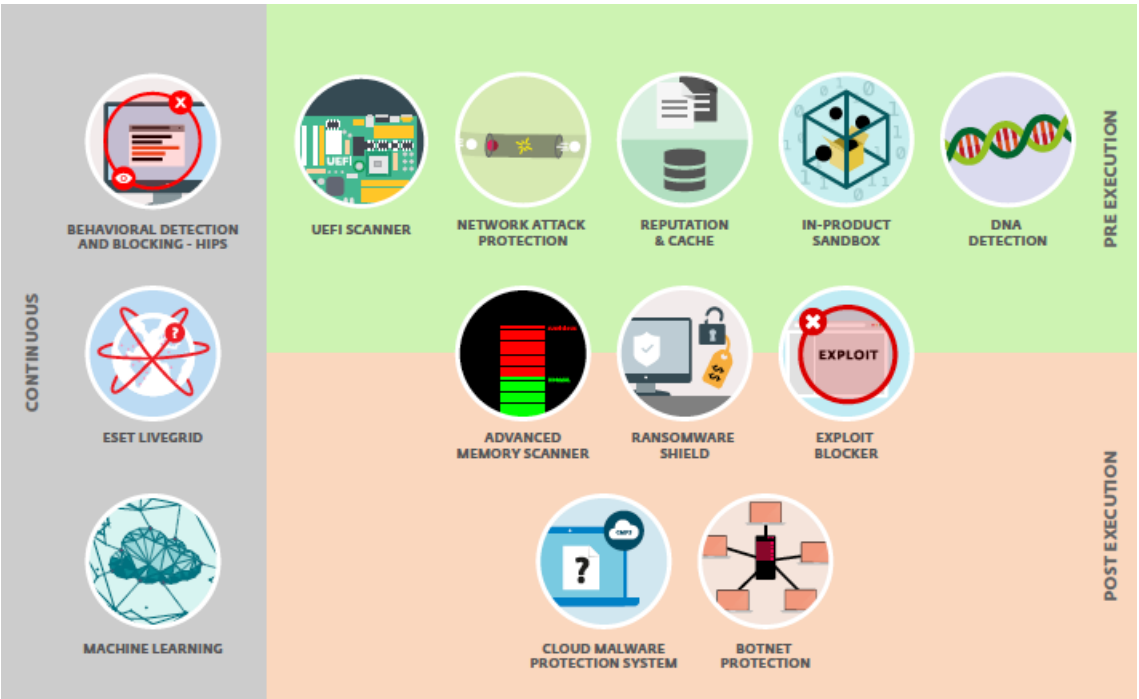
All patches are updates, but not all updates are patches. Whereas patches are used within the context of fixing something specific, security updates are implemented for general security purposes rather than, for example, targeting a particular type of malware or vulnerability.

Patches can help stop malware from spreading, but they are not guaranteed to prevent or halt a cyberattack. One of the main challenges firms often experience during an attack is that they lack the capacity to apply patches quickly to a high number of machines, because they cannot allow their machines to be out of action for a prolonged amount of time.

Many exploits take advantage of known vulnerabilities that may have already been patched, so update and patch your operating system and applications.

Protection layers

Established security companies, have maintained their market share by evolving to address ever changing threats. Today's threats cannot be fought effectively by just building on technology from the 1990s. Fighting modern malware is a cat-and-mouse game in which the industry faces teams of skilled and (financially) motivated cybercriminals and state actors. So security companies need to refine their products constantly, both reactively and proactively, to provide effective solutions, adding different layers by which modern malware can be detected and/or blocked. A single point of protection or a single method of defense is simply not enough.



The figure on the previous page shows how ESET's various core technologies are layered and an approximation of when and how they can detect and/or block a threat during its lifecycle in the system. This visualization should assist in understanding where the technologies described below fit into a layered security architecture.

Firmware is the link between your hardware and software, it tells your device how to boot. If malware is present prior to the operating system starting then it is infected before it becomes operational. Scanning firmware and checking whether it is free from infection ensures the security of the pre-boot environment is compliant with the firmware specifications.

Machine Learning may inspire images of George Orwell's 1984 and machines taking over the world. However, between 2006 and 2007 there was a massive increase in the number of malware samples seen in the real world. The number of samples in the last 12 years has gone from single digit millions to over 700 million today. The use of machine learning allows for the automation of decision making on whether a sample is malicious, freeing up human resources to research and understand more sophisticated attacks.

The majority of security vendors have adopted machine learning into both their infrastructure and directly into their products. When a vendor has vast networks of client machines reporting and analyzing file access and communications the data from this network can be used to teach a machine learning algorithm the difference between good and malicious code. In simple terms this is about training a machine to solve a problem, if I want the machine to identify the difference between a football and a tennis ball then I need to show it examples. By feeding in pictures of maybe 100 of each type of ball the machine will then have enough data to make a decision on which is which. The importance lies in having enough data to make the correct decision, especially when determining if something is malicious, or not.

The internet works at lightning speeds, a malware infection that starts in Asia in the afternoon can result in a mass infection in Europe a short time later. When discussing machine learning above we mentioned that the collective intelligence of large numbers of machines to be important. When malicious code is detected and that intelligence is shared with a cloud-based detection and reputation system then the entire community of devices using the data become protected. It is of course important to be sure of the decision on whether or not something is malicious, and then use intelligence from multiple layers to determine whether to share a decision through the cloud, because blocking a good file, more commonly known as a false positive, can be as devastating to a business as not detecting a malicious one.

A botnet is a network of private computers that are infected with malicious software. The devices are controlled as a group and provide cybercriminals extensive resources that would otherwise be unavailable. Typical attacks launched using botnets are denial of service attacks or the sending of spam messages. For a botnet to be effective it needs to communicate, so detecting the malicious communication used by botnets and identifying the processes being used can result in the shut down and removal of the malware.

When a threat, such as ransomware has specific characteristics then it is common to find a layer in the security model that looks for the common attack mechanisms used. For example ransomware will attempt to access the file system, display messages, or is delivered in a certain way. By monitoring and evaluating all executed applications based on their behavior and reputation a decision to detect and block processes that resemble the behavior of ransomware can be made.

Today's malware is often heavily obfuscated and tries to evade detection as much as possible. To see through this and identify the real behavior hidden underneath the surface, in-product

sandboxing can be used. The emulation of different components of computer hardware and software allow the execution of a suspicious sample in an isolated virtualized environment. Watching and monitoring the behavior for expected results, or in the case of malware, unexpected results, allows the system to either warn the user of abnormalities or to make a policy decision and block the execution in the real environment.

A Host-based Intrusion Prevention System (HIPS) is a process that monitors system activity and uses a pre-defined set of rules and data to recognize suspicious system behavior. By analyzing system calls, application logs, file-system modifications and the use of memory, intrusions can be detected. When this type of activity is identified, the HIPS self-defense mechanism stops the offending program or process from carrying out potentially harmful activity.

There are of course other layers and technologies in use, as you can see from the ESET visualization. The technologies above describe some of the layers in modern security products. It is not an exhaustive list but more a demonstration that when deciding on an approach to protect a network it is important to look at all aspects of protection and the different advantages each add to a solution.

No white paper like this one would be complete without reiterating the fundamental security protection tips below:

1. Use a reliable security solution that employs multiple layers to protect you from threats.
2. Keep backups on an offline hard disk or location that will not be hit in case of a network infection.
3. In the case of Ransomware – do not pay. There is no guarantee that paying will lead to decryption. There have been cases of no decryptor or key being sent after the payment was made and others where buggy code means files cannot be properly decrypted anyway. And funding criminals only encourages their malicious behavior.
4. Ensure your network is wellconfigured and segmented, and constantly monitor traffic for any abnormal behavior.
5. It is essential to manage passwords carefully. If the same password is used across different management centers, even if only one of the infected machines possesses the credentials of administrator, this could infect the whole network.
6. Use two-factor authentication, since it adds an additional layer of protection to the credentials normally used for validating users. In case of infection, this prevents lateral movement across your network should the malware try to gain remote administrative access to other computers.
7. Educate employees to delete suspicious emails, instead of opening the attachments, or clicking on the links that take them to phony sites that steal credentials or deliver malware.