Hunting for persistence via Microsoft Exchange Server or Outlook

Microsoft Exchange and Outlook are sufficient parts of almost any corporate infrastructure, regardless of its size. MS Exchange Servers are desired target for attackers, since in case of successful exploitation of vulnerabilities or incorrect settings of MS Exchange components, attackers can gain access to emails of the company, increase their privileges to the domain administrator, and also perform phishing mailing on behalf of the organization's representatives. Moreover, attackers have a number of original ways to obtain persistence in the system. The speaker will consider all these methods and demonstrate approaches to obtain persistence using these methods and detect such presence.

Heirhabarov
May 21, 2021

More Decks by Heirhabarov

See All by Heirhabarov >

Hunting For The Most Unusual Attack Te...

heirhabarov ☆ 1 👁 170

Hunting for macOS attack techniques. Pa...

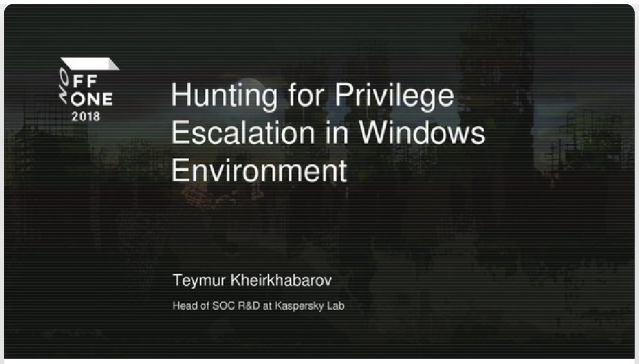
heirhabarov ☆ 2 👁 2.7k

Hunting for Active Directory Certificate ...

heirhabarov ☆ 2 👁 6.5k

Hunting for PowerShell Abuse

heirhabarov ☆ 9 👁 18k



Hunting for Privilege Escalation in Windo...

heirhabarov ☆ 13 👁 28k

PHDays 2018 Threat Hunting Hands-On ...

heirhabarov ☆ 7 👁 5.2k

Hunting for Credentials Dumping in Win...

heirhabarov ☆ 5 👁 6k

Other Decks in Technology

See All in Technology >

TinyMLの技術動向

kyotomon ☆ 2 👁 270

30万人が利用するチャットをFirebase R...

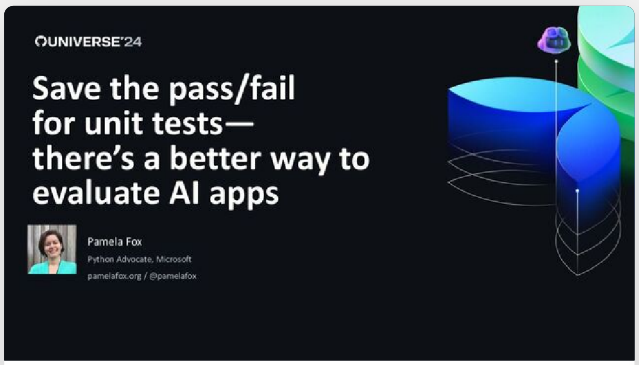
ryosk7 ☆ 2 👁 300

Figma Dev Modeで進化するデザインと...

cyberagentdevelop... ☆ 1 👁 380

Nix入門パラダイム編

asa1984 ☆ 1 👁 170



GitHub Universe: Evaluating RAG apps in...

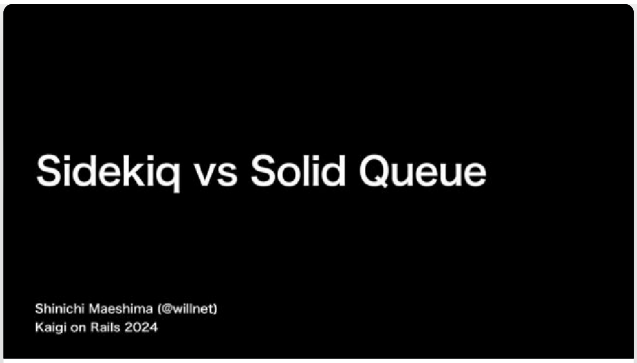
pamelafox ☆ 0 👁 140

WINTICKETアプリで実現した高可用性...

cyberagentdevelop... ☆ 1 👁 180

なんで、私がAWS Heroに!? ～社外の広...

minorun365 PRO ☆ 4 👁 780



Sidekiq vs Solid Queue

willnet ☆ 11 👁 7.1k

AIを使って小説を書こう！【2024/10/25...

kamomeashizawa ☆ 0 👁 170

現地でMeet Upをやる場合の注意点⍰ー...

shotashiratori ☆ 0 👁 240

新卒1年目が挑む！生成AI×マルチエー...

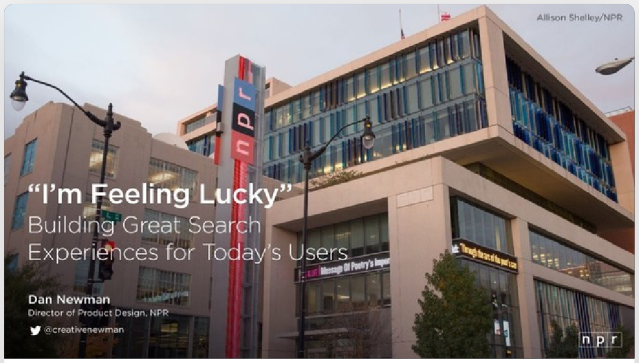
cyberagentdevelop... ☆ 0 👁 110

[PyCon Korea 2024] Lightning Talk: PyPI...

studioego PRO ☆ 0 👁 130

Featured

See All Featured >



"I'm Feeling Lucky" - Building Great Sear...

danielanewman ☆ 225 👁 22k

Principles of Awesome APIs and How to ...

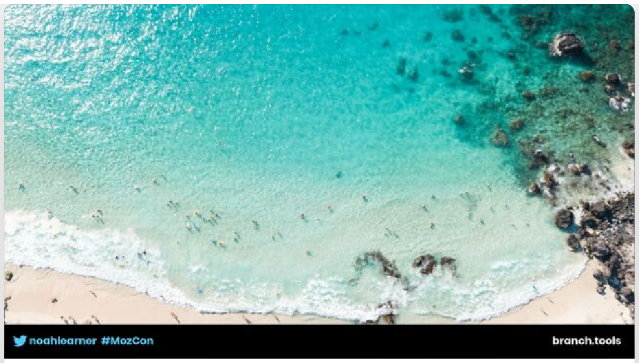
keavy ☆ 126 👁 17k

Docker and Python

trallard ☆ 40 👁 3.1k

No one is an island. Learnings from foster...

thoeni ☆ 19 👁 3k



Into the Great Unknown - MozCon

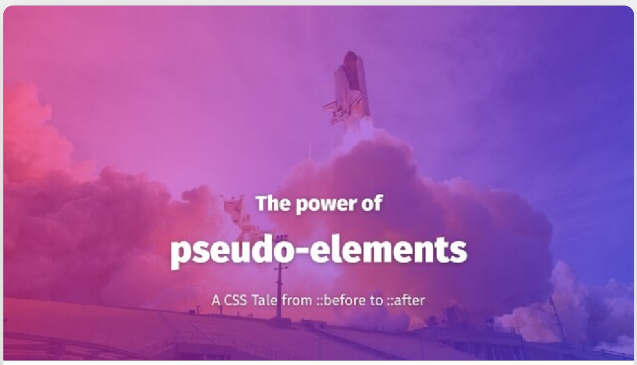
thekraken ☆ 31 👁 1.5k

Dealing with People You Can't Stand - Big...

cassininazir ☆ 364 👁 22k

Responsive Adventures: Dirty Tricks Fro...

smashingmag ☆ 250 👁 21k

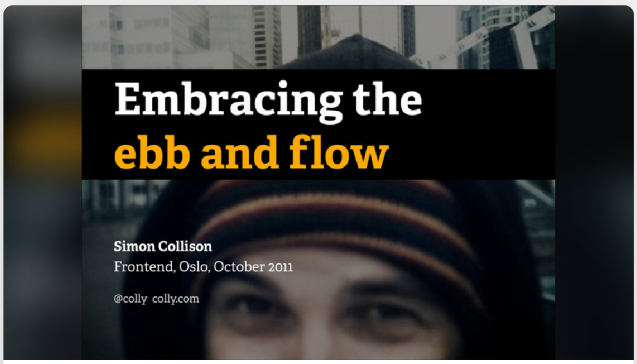


The Power of CSS Pseudo Elements

geoffreycrofte ☆ 72 👁 5.3k

Teambox: Starting and Learning

jrom ☆ 132 👁 8.7k

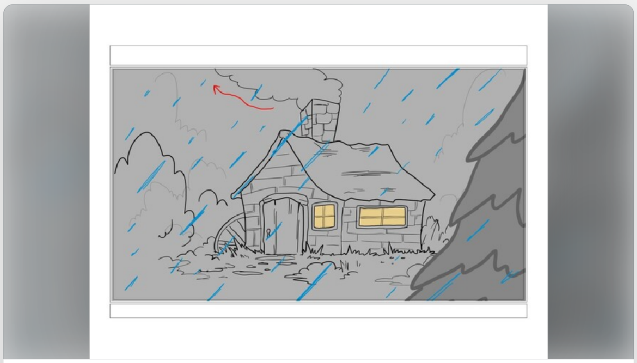


Embracing the Ebb and Flow

colly ☆ 84 👁 4.4k

YesSQL, Process and Tooling at Scale

rocio ☆ 167 👁 14k



Fireside Chat

paigeccino ☆ 32 👁 3k

Transcript

1. **1 Hunting for persistence via Exchange and Outlook capabilities Teymur**

   Kheirkhabarov Head of SOC, BI.ZONE Anton Medvedev Principal SOC Analyst, BI.ZONE

2. **2 Who we are? • Head of SOC at BI.ZONE**

   • Threat Hunter • ZeroNights / PHDays / OFFZONE speaker • GIAC GXPN / GCFA / GDSA certified • Ex- Head of SOC R&D at Kaspersky Lab / SOC Analyst Infosec Admin/Engineer • Twitter @HeirhabarovT • heirhabarov@gmail.com • Principal SOC Analyst at BI.ZONE • Threat Hunter • OSCP certified • Twitter @BigToni94 • medvedevanton23@gmail.com Anton Medvedev Teymur Kheirkhabarov

3. **3 Persistence via Exchange and Outlook capabilities Exchange server side**

4. **4 Exchange Transport Agent Transport agents let you install custom**

   software on an Exchange server which can then process email messages that pass through the transport pipeline to perform various tasks such as filtering spam, filtering malicious attachments, journaling, or adding a corporate signature to the end of all outgoing emails. The Microsoft Exchange Server Transport Agents SDK allows third parties to implement the following predefined classes of transport agents: • SmtpReceiveAgent • RoutingAgent • DeliveryAgent Transport agents use SMTP events. These events are triggered as messages move through the transport pipeline. SMTP events give transport agents access to messages at specific points during the SMTP conversation and during routing of messages through the organization. Transport agents have full access to all e-mail messages that they encounter. Exchange puts no restrictions on a transport agent's behavior.

5. **5 Abusing Exchange Transport Agent T1505.002 – Server Software Component:**

   Transport Agent Adversaries may register a malicious transport agent to provide a persistence mechanism in Exchange Server that can be triggered by adversary-specified email events. Though a malicious transport agent may be invoked for all emails passing through the Exchange transport pipeline, the agent can be configured to only carry out specific tasks in response to adversary defined criteria. LightNeuron – Turla's backdoor specifically designed to target Microsoft Exchange mail servers. LightNeuron is the first publicly known malware to use a malicious Microsoft Exchange Transport Agent. It allows to perform the next operations: • Modify emails; • Block emails; • Create new emails; • Dump emails and attachments; • Execute arbitrary .NET assembly code on the Exchange server. https://www.welivesecurity.com/wp-content/uploads/2019/05/ESET-LightNeuron.pdf

6. **6 Exchange Transport Agent Example An example of a Transport**

   Agent that, upon receipt of any message, launches cmd.exe and add "– Evil Agent Subject" to the subject of the received mail

7. **7 Exchange Transport Agent Installation Artifacts The PowerShell cmdlets "Install-TransportAgent"**

   and "Enable-TransportAgent" can be used to register and activate transport agents on Exchange servers.

8. **8 Exchange Transport Agent Installation Artifacts**

9. **9 Exchange Transport Agent Installation Artifacts Let's hunt it! Search**

   for usage of "Install-TransportAgent" and "Enable-TransportAgent" cmdlets in the "MSExchange Management" event log: Channel:"MSExchange Management" AND SourceName:"MSExchange CmdletLogs" AND EventID:(1 OR 6) AND Message:("*Install-TransportAgent*" OR "*Enable-TransportAgent*")

10. **10 Exchange Transport Agent Installation Artifacts It is also possible**

    to find the signs of usage "Install-TransportAgent" and "Enable-TransportAgent" in the PowerShell events log ("Windows PowerShell" and "Microsoft-Windows-PowerShell/Operational"):

11. **11 Exchange Transport Agent Installation Artifacts Let's hunt it! Search**

    for usage of "Install-TransportAgent" and "Enable-TransportAgent" cmdlets in the PowerShell event logs: (Channel:"Microsoft-Windows-PowerShell/Operational" AND EventID:4104 AND ScriptBlockText.keyword:(Enable- TransportAgent* OR Install-TransportAgent*)) OR (Channel:"Windows PowerShell" AND EventID:800 AND Message:("*Enable-TransportAgent*" OR "*Install-TransportAgent*"))

12. **12 Exchange Transport Agent Configuration File Transport Agent management cmdlets**

    manipulate the configuration file agents.config located at %ExchangeInstallPath%\TransportRoles\Shared. In order to hide his activity, an adversary can directly modify this file without usage of any PowerShell cmdlets.

13. **13 Exchange Transport Agent Configuration File Change**
14. **14 Exchange Transport Agent Configuration File Change Let's hunt it!**

    Search for Exchange Transport Agent configuration file changes: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:11 AND TargetFilename:"*\\TransportRoles\\Shared\\agents.config" AND -Image:"*\\ExchangeSetup\\ExSetupUI.exe"
15. **15 Exchange Transport Agent Loading Transport Agent – it isn't**

    DLL, it is .NET assembly, that is loaded by EdgeTransport.exe process
16. **16 Exchange Transport Agent Loading Module loading event from Microsoft-Windows-**

    DotNETRuntime ETW provider
17. **17 Spawning new process via Exchange Transport Agent Spawning new**

    process with EdgeTransport.exe as a parent
18. **18 Spawning new process via Exchange Transport Agent Let's hunt**

    it! Search for spawning new process with EdgeTransport.exe as a parent: (Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1 AND ParentImage:"*\\Bin\\EdgeTransport.exe" AND -Image:"\\Bin\\OleConverter.exe") OR (Channel:"Security" AND EventID:4688 AND ParentProcessName:"*\\Bin\\EdgeTransport.exe" AND -NewProcessName:"\\Bin\\OleConverter.exe")
19. **19 IIS Extensions T1505 – Server Software Component ISAPI Filters**

    ISAPI filters are DLL files that can be used to modify and enhance the functionality provided by IIS. ISAPI filters always run on an IIS server, filtering every request until they find one they need to process. The ability to examine and modify both incoming and outgoing streams of data makes ISAPI filters powerful and flexible. Managed-code/Native-code HTTP modules IIS 7.0 and above have been re-engineered from the ground up to provide a brand new C++ and .NET APIs, on which all of the in-the-box features are based, to allow complete runtime extensibility of the web server. HTTP modules are based on this new architecture. An HTTP module is called on every request that is made to your application. HTTP modules are called as part of the ASP.NET request pipeline and have access to life- cycle events throughout the request. HTTP modules let you examine incoming and outgoing requests and take action based on the request. IIS extensions can be used: change request data sent by the client, modify a response going back to the client, run processing when a request is complete, perform special logging or traffic analysis, perform custom authentication, etc.
20. **20 IIS Managed-code HTTP module example**
21. **21 IIS HTTP Module installation using AppCmd In order to**

    install a module/ISAPI filter, it must be registered with the server using one of the options below: • Using the IIS Manager • Using the AppCmd.exe command line tool • Manually editing the IIS configuration file

22. **22 IIS ISAPI Filter installation using AppCmd**
23. **23 IIS ISAPI Filter/HTTP Module installation using AppCmd Let's hunt**

    it! Search for the appcmd.exe process creation with specific command line: ((Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1) OR (Channel:Security AND EventID:4688)) AND CommandLine:*appcmd* AND (CommandLine:"*add module*" OR CommandLine:("*set config*" AND *isapiFilters*))
24. **24 IIS ISAPI Filter/HTTP Module installation using IIS Manager**
25. **26 IIS ISAPI Filter/HTTP Module installation via config editing Server**

    Level – applicationHost.config file
26. **27 IIS ISAPI Filter/HTTP Module installation via config editing Site**

    Level – appropriate Web.config file
27. **28 IIS ISAPI Filter/HTTP Module installation via config editing**
28. **29 IIS ISAPI Filter/HTTP Module installation via config editing Let's**

    hunt it! Search for creation/modification of the web.config or applicationHost.config files: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:11 AND TargetFilename: ("*\\inetsrv\\config\\applicationHost.config" OR "*\\web.config")
29. **30 IIS ISAPI Filter/Native-code HTTP Module Loading**
30. **31 IIS Managed-code HTTP Module Loading**
31. **32 Web Shell T1505.003 – Server Software Component: Shell Recently,**

    a large number of incidents have been occurred in which the use of web shell observed in post-compromised Microsoft Exchange Servers. After successful exploiting a Microsoft Exchange Server vulnerability for initial accesses, an adversary can upload a web shell to enable remote administration of the affected system.
32. **33 Spawning new process via Web Shell Spawning new process**

    by IIS worker process (w3wp.exe)
33. **34 Spawning new process via Web Shell / IIS Extension**

    Let's hunt it! Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1 AND ParentImage:"\\w3wp.exe" AND (CommandLine:(cmd.exe OR "*cmd *" OR *comspec* OR *wscript* OR *cscript* OR *SyncAppvPublishingServer* OR *powershell* *pwsh*) OR OriginalFileName:("cmd.exe" OR "wscript.exe" OR "cscript.exe" OR "SyncAppvPublishingServer.exe" OR "PowerShell.EXE") OR Image:"\\cmd.exe") AND -CommandLine:("cmd /C exit" OR "cmd.exe /c set") Channel:Security AND EventID:4688 AND ParentProcessName:"\\w3wp.exe" AND (CommandLine:(cmd.exe OR "*cmd *" OR *comspec* OR *wscript* OR *cscript* OR *SyncAppvPublishingServer* OR *powershell* *pwsh*) OR Image:"\\cmd.exe") AND -CommandLine:("cmd /C exit" OR "cmd.exe /c set") Search for spawning suspicious processes (PowerShell, cmd, wscript, cscript) by IIS worker process (w3wp.exe):

34. **35 Dropping files that appear to be the web shells**

    Search for dropping files that appear to be the web shells, including dropping via SMB:

35. **36 Dropping files that appear to be the web shells**

    Let's hunt it! Search for dropping files that appear to be the web shells: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:11 AND TargetFilename.keyword:(*.pht OR *.phtml OR *.php OR *.php1 OR *.php2 OR *.php3 OR *.php4 OR *.php5 OR *.php6 OR *.php7 OR *.asp OR *.aspx OR *.ashx OR *.aspq OR *.axd OR *.cshtm OR *.cshtml OR *.vbhtm OR *.vbhtml OR *.asa OR *.shtml OR *.jsp OR *.jspx OR *.war) AND TargetFilename:("*\\ClientAccess\\Owa*" OR "*\\HttpProxy\\Owa*" OR "*\\www*" OR "*\\html*" OR "*\\htdocs*" OR "*\\inetpub\\wwwroot*" OR "*\\microsoft shared\\web server extension*" OR "*\\ClientAccess\\ecp*" OR "*\\HttpProxy\\ecp*")

36. **37 Search for dropping files via SMB share that appear**

    to be the web shells: Channel:Security AND EventID:5145 AND AccessList:(*4417* OR *4418*) AND RelativeTargetName.keyword:(*.pht OR *.phtml OR *.php OR *.php1 OR *.php2 OR *.php3 OR *.php4 OR *.php5 OR *.php6 OR *.php7 OR *.asp OR *.aspx OR *.ashx OR *.aspq OR *.axd OR *.cshtm OR *.cshtml OR *.vbhtm OR *.vbhtml OR *.asa OR *.shtml OR *.jsp OR *.jspx OR *.war) AND RelativeTargetName:("*\\ClientAccess\\Owa*" OR "*\\HttpProxy\\Owa*" OR "*\\www*" OR "*\\html*" OR "*\\htdocs*" OR "*\\inetpub\\wwwroot*" OR "*\\microsoft shared\\web server extension*" OR "*\\ClientAccess\\ecp*" OR "*\\HttpProxy\\ecp*") %%4417 – WriteData (or AddFile) %%4418 – AppendData (or AddSubdirectory or CreatePipeInstance) Dropping files that appear to be the web shells Let's hunt it!

37. **38 Persistence via Microsoft Exchange Server and Outlook Outlook client**

    side

38. **39 Ruler Tool Ruler is a tool that allows to**

    interact with Exchange servers remotely, through either the MAPI/HTTP or RPC/HTTP protocol. The main aim is abuse the client-side Outlook features and gain a shell remotely. Ruler has multiple functions: • enumerate valid users; • dump the Global Address List (GAL); • create new malicious mail rules; • VBScript execution through forms; • VBScript execution through the Outlook Home Page. https://github.com/sensepost/ruler

39. **40 Ruler tool artifacts – hardcoded workstation name Ruler uses**

    hardcoded workstation name for the logon to the Windows hosts – "RULER".

40. **41 Ruler tool artifacts – hardcoded workstation name Let's hunt**

    it! Search for 4624, 4625 or 4776 events, where workstation name is "RULER": (Channel:Security AND EventID:(4776) AND Workstation:RULER) OR (Channel:Security AND EventID:(4625 OR 4624) AND WorkstationName:RULER)

41. **42 Ruler tool artifacts – hardcoded user-agent 2021-05-14 11:46:57 10.3.132.20**

    GET /autodiscover/autodiscover.xml &CorrelationID=<empty>;&cafeReqId=67952f2e-d8a3-44ad-9a60- 898d36c8192c; 443 - 172.21.194.203 ruler - 1 2148074254 10 2021-05-14 11:46:57 10.3.132.20 POST /autodiscover/autodiscover.xml &CorrelationID= <empty>;&cafeReqId=b3d7ef4f-06fb-4092-856a- d4af9dd155b7; 443 LAB\user1 172.21.194.203 ruler - 0 0 270 2021-05-14 11:46:57 10.3.132.20 POST /mapi/emsmdb/ MailboxId=43873a7d-0aac-45e5-b531- d7f7bbf82d32@lab.local&CorrelationID= <empty>;&ClientRequestInfo= R:{C715155F-2BE8-44E0-BD34-2960065754C8}:2;RT:Connect;CI:{2F94A2BF-A2E6-4CCC-BF98- B5F22C542226};CID: <null>&cafeReqId=ee775774-5f7d-4389-8695- 7dbba95ed42c; 443 LAB\user1 172.21.194.203 ruler - 0 0 132

42. **43 Outlook Rules T1137.005 – Office Application Startup: Outlook Rules**

    Outlook rules allow a user to define automated behavior to manage email messages. A benign rule might, for example, automatically move an email to a particular folder in Outlook if it contains specific words from a specific sender. Adversaries may abuse Microsoft Outlook rules to obtain persistence on a compromised system. Malicious Outlook rules can be created that can trigger code execution when an adversary sends a specifically crafted email to that user. It can be achieved via “Start application” and “Run a script” rule actions. Once malicious rules have been added to the user’s mailbox, they will be loaded when Outlook is started. Malicious rules will execute when an adversary sends a specifically crafted email to the user.

43. **44 Outlook Rules Fully updated and patched versions of Outlook**

    2013, and 2016 disable the “Start application” and “Run a script” rule actions by default. This will ensure that even if an attacker breaches the account, the rule actions will be blocked. Here are the patch versions for your Outlook 2013 and 2016 clients: • Outlook 2016: 16.0.4534.1001 or greater; • Outlook 2013: 15.0.4937.1000 or greater. There are no ”Start application” and “Run a script” rule actions

44. **45 Enable unsafe Outlook Rules To re-enable “Start application” and**

    “Run a script” rule actions, you can create and set the EnableUnsafeClientMailRules Registry value: • Key: HKEY_CURRENT_USER\Software\Microsoft\Office\<ver sion>\Outlook\Security • Value name: EnableUnsafeClientMailRules • Value type: REG_DWORD • Value: 1

45. **46 Enable unsafe Outlook Rules**

46. **47 Enable unsafe Outlook Rules Let’s hunt it! Search for**

    modification of EnableUnsafeClientMailRules Registry value: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13 AND TargetObject:"\\EnableUnsafeClientMailRules" AND Details:"DWORD (0x00000001)" Search for usage of standard Windows tools to create and set the EnableUnsafeClientMailRules Registry value: ((Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1) OR (Channel:Security AND EventID:4688)) AND CommandLine:*EnableUnsafeClientMailRules*

47. **48 Uses Ruler to create malicious Outlook Rules Outlook uses**

    ShellExec to open the payload application which means that the payload can't be executed with arguments, requiring the payload to be an all enclosed application hosted on the disk or externally (via SMB or WebDav). Externally hosted payload is the most common and reliable way for an adversary who is going to use Outlook Rules.

48. **49 Is it possible to detect creation of Rules on**

    the server side? The answer is unfortunately no! The Exchange server logs don't contain any significant event for the detection. RPC event: 2021-05-14T12:43:34.255Z,EXCHANGE,RpcHttp,S:Stage=EndRequest;S:UserName=LAB\user1;S:AuthType=NTLM;S:Status=200.0.OK;S:HttpVerb =RPC_IN_DATA;S:UriQueryString=?43873a7d-0aac-45e5-b531-d7f7bbf82d32@lab.local:6001;S:RequestId=8c2f7c07-11db-4ff6-838a-f84b61a8aea4;S:ClientIp= 172.21.194.203 MAPI event: 2021-05-14T12:16:28.480Z,1a5792de-6350-4d18-8259-067a2d465f29,{C715155F-2BE8-44E0-BD34- 2960065754C8}:3, <null>,Execute,200,0,0,0,27,Unknown,15,1,1591,10,LAB\user1,,,,,43873a7d-0aac-45e5-b531-d7f7bbf82d32@lab.local,9a179873-e3e7-4408-838b-54fb489dbd2c,user1@lab.local,172.21.194.203,EXCHANGE.LAB.LOCAL, <null>,,MAPIAAAAAOC4+7PyvPu+na+frZyxgbSZqJy8jLuBtYK4jL njwPHB8Mj6yf/L/M7JAQAAAAAAAA==,0-5QcQfg==,{2F94A2BF-A2E6-4CCC-BF98-B5F22C542226},,15.0.4815.1002,0,Negotiate,,,,,,,,,Anonymous,>[254] <[254],OwnerLogon;LogonId: 12;,cpn=M_ABR/RUM_ABR/RUM_ABRC/M_APAR/M_APRH/M_DTC/M_DTQ/M_DTE/M_RDE/M_RDrE/M_RDrEc/M_RDEc/M_DTEc/ M_APoRH/M_AER/;cpv=0/2/2/4/4/6/6/6/6/7/26/26/26/28/28/;Dbl:ST.T[exchange.9a179873-e3e7-4408-838b-54fb489dbd2c]=1;Dbl:BudgUse.T[]=38.002799987793;Dbl:MAPI.T[exchange.9a179873-e3e7-4408-838b- 54fb489dbd2c]=7;Dbl:EXR.T[exchange.9a179873-e3e7-4408-838b-54fb489dbd2c]=3;Dbl:VCGS.T[EXCHANGE]=1;I32:VCGS.C[EXCHANGE]=1;I32:ROP.C[exchange.9a179873-e3e7-4408-838b- 54fb489dbd2c]=1634283;I32:MAPI.C[exchange.9a179873-e3e7-4408-838b-54fb489dbd2c]=40;I32:RPC.C[exchange.9a179873-e3e7- 4408-838b-54fb489dbd2c]=3;Dbl:RPC.T[exchange.9a179873-e3e7-4408-838b-54fb489dbd2c]=6;I32:MB.C[exchange.9a179873-e3e7- 4408-838b-54fb489dbd2c]=3;F:MB.AL[exchange.9a179873-e3e7-4408-838b-54fb489dbd2c]=2,

49. **50 Launching a remote payloads using Outlook Rules Windows /**

    Sysmon events, related to the execution of a binary from remote SMB/WebDav share by outlook.exe process:

50. **51 Search for process creation events where outlook.exe is a**

    parent and started executable is located on SMB or WebDav share: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1 AND ParentImage:"\\outlook.exe" AND Image.keyword:/\\\\.+\\.+/ Channel:Security AND EventID:4688 AND ParentProcessName:"\\outlook.exe" AND NewProcessName.keyword:/\\Device\\Mup\\.+\\.+/ Launching a remote payloads using Outlook Rules Let's hunt it!

51. **52 Outlook macro-based persistence T1137.001 – Office Application Startup: Office**

    Template Macros Microsoft Outlook stores Microsoft Visual Basic for Applications (VBA) code in a file that's named VbaProject.OTM (<Drive>:\Users\<LogonName>\ AppData\Roaming\Microsoft\Outlook\VbaProject.OTM). Unlike other Microsoft Office programs, Outlook supports only one VBA project at a time. Attacker can replace or modify this file in order to inject visual basic code that will execute each time an Outlook starts or in case of other events (for example, in case of receiving new email).

52. **53 Launching VBScript from Outlook Macros using Rules It is**

    also possible to use rules for running visual basic code from the Outlook macros file VbaProject.OTM. Function name from the VbaProject.OTM should be specified as a rule parameter:

53. **54 The attackers replaced Outlook's original VbaProject.OTM file with a**

    malicious macro that serves as the backdoor. The backdoor receives commands from a Gmail address operated by the threat actor, executes them on the compromised machines and sends the requested information to the attacker's Gmail account. Before the attackers deployed the macro-based backdoor, they had to take care of two things: • Creating persistence - the attackers modified specific registry values to create persistence: https://www.cybereason.com /hubfs/Cybereason%20Labs% 20Analysis%20Operation%20C obalt%20Kitty-Part2.pdf Outlook macro-based persistence REG ADD "HKCU\Software\Microsoft\Office\14\Outlook" /v "LoadMacroProviderOnBoot" /f /t REG_DWORD /d 1 • Disabling Outlook's security policies - to do that, the attackers modified security settings to enable the macro to run without prompting any warnings to the users: Finally, the attackers replaced the existing VbaProject.OTM with the fake macro: REG ADD "HKCU\Software\Microsoft\Office\14\Outlook\Security" /v "Level" /f /t REG_DWORD /d 1 cmd /c cd c:\programdata\& copy VbaProject.OTM C:\Users\ [REDACTED]\AppData\ Roaming\Microsoft\Outlook

54. **55 Outlook macro-based persistence**

55. **56 Outlook Macros security settings change Let's hunt it! Search**

    for modification of the Outlook's security settings to enable Outlook macro-based persistence: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13 AND TargetObject: ("*\\Outlook\\Security\\Level" OR "*\\Outlook\\LoadMacroProviderOnBoot") AND Details:"DWORD (0x00000001)"

56. **57 Outlook Macros security settings change Let's hunt it! Search**

    for usage of standard Windows tools for modification of the Outlook's security settings to enable Outlook macro-based persistence: ((Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1) OR (Channel:Security AND EventID:4688)) AND ((CommandLine:"*\\Outlook\\Security" AND CommandLine:*Level*) OR (CommandLine:"*\\Outlook" AND CommandLine:*LoadMacroProviderOnBoot*))

57. **58 Outlook Macros file modification/replacement Let's hunt it! Search for**

    unauthorized modification or replacement of VbaProject.OTM file: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:11 AND TargetFilename:"*\\AppData\\Roaming\\Microsoft\\Outlook\\VbaProject.OTM"

58. **59 Loading of the "Microsoft VBA for Outlook Addin" Loading**

    of the "Microsoft VBA for Outlook Addin" (OUTLVBA.DLL) by the Outlook process can be the sign of VBScript code execution from an Outlook Macros file (VbaProject.OTM)

59. **60 Loading of the "Microsoft VBA for Outlook Addin" Let's**

    hunt it! Search for loading of the "Microsoft VBA for Outlook Addin": Channel:Application AND SourceName:Outlook AND EventID:45 AND Message:"*Microsoft VBA for Outlook Addin*" Search for loading of the "OUTLVBA.DLL" library by Outlook process: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:7 AND ImageLoaded:"\\ADDINS\\OUTLVBA.DLL" AND Image:"*\\OUTLOOK.EXE"

60. **61 Outlook Home Page T1137.004 – Office Application Startup: Outlook**

    Home Page Outlook Home Page is a legacy feature used to customize the presentation of Outlook folders. This feature allows for an internal or external URL to be loaded and presented whenever a folder is opened. Adversaries may abuse Microsoft Outlook's Home Page feature to obtain persistence on a compromised system by creating a malicious HTML page. Once malicious home pages have been added to the user's mailbox, they will be loaded when Outlook is started. Malicious Home Pages will execute when the right Outlook folder is loaded/reloaded.

61. **62 Outlook Home Page example Outlook Home Page allows to**

    execute arbitrary visual basic script, that's why it is so attractive for adversaries:

62. **63 Outlook Home Page "protection" On October 10, 2017, Microsoft**

    released patches for Microsoft Outlook to "protect" against home page abusing: • KB4011196 (Outlook 2010) • KB4011178 (Outlook 2013) • KB4011162 (Outlook 2016) These patches just make it impossible to set a home page URL from the Outlook user interface (UI) by hiding the "Home Page" tab in the folder properties. But it is possible for an attacker to re-enable the original home page tab by creating and set the specific Registry value. There is no "Home Page" tab in the folder properties UI

63. **64 Enable home page tab in the Outlook UI To**

    re-enable the original home page tab and roaming home page behavior in the Outlook UI, you can create and set the EnableRoamingFolderHomepages Registry value: The following setting will allow for folders within secondary (non- default) mailboxes to leverage a custom home page. HKCU\Software\Microsoft\Office\<version>\Outlook\Security “EnableRoamingFolderHomepages” = REG_DWORD:00000001 HKCU\Software\Microsoft\Office\<version>\Outlook\Security “NonDefaultStoreScript” = REG_DWORD:00000001 And now we have home page tab in the UI

64. **65 Enable home page tab in the Outlook UI**

65. **66 Enable home page tab in the Outlook UI Let’s**

    hunt it! Search for modification of EnableRoamingFolderHomepages and NonDefaultStoreScript Registry value: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13 AND TargetObject:(“\\EnableRoamingFolderHomepages” OR “\\NonDefaultStoreScript”) AND Details:"DWORD (0x00000001)" Search for usage of standard Windows tools to create and set the EnableRoamingFolderHomepages and NonDefaultStoreScript Registry values: ((Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1) OR (Channel:Security AND EventID:4688)) AND CommandLine:(*EnableUnsafeClientMailRules* OR *NonDefaultStoreScript*)

66. **67 Enable home page tab in the Outlook UI. Registry**

    artifact When you change home page URL for folder via UI some interesting Registry values are created and changed under HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Webcheck\Store.?\{GUID}:

67. **68 Enable home page tab in the Outlook UI. Registry**

    artifact Let’s hunt it! Search for modification of the “(Defualt)” Registry value under HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Webcheck\Store.?\{GUID} (Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13) AND TargetObject:"*\\Microsoft\\Windows\\CurrentVersion\\Webcheck*" AND Details:Outlook*

68. **69 Set Outlook Home Page via Registry The FireEye Red**

    Team found that an attacker can set a home page to achieve code execution and persistence by editing the specific registry keys. Setting this registry key to a valid URL enables the home page regardless of the patch being applied or not. Although the option will not be accessible from the Outlook user interface (UI), it will still be set and render These keys are set within the logged-on user’s Registry hive. This means that no special privileges are required to edit the Registry and roll back the patch. The FireEye Red Team found that no other registry modifications were required to set a malicious Outlook homepage.

69. **70 Set Outlook Home Page via Registry example https://www.fireeye.com/blog/threat-research/2019/12/breaking-the-rules-**

    tough-outlook-for-home-page-attacks.html The FireEye Advanced Practices team discovered a uniquely automated phishing document was uploaded to VirusTotal. The sample, “TARA Pipeline.xlsm” (MD5: ddbc153e4e63f7b8b6f7aa10a8fad514), launches malicious Excel macros combining several techniques, including using the lesser-known HKCU\Software\Microsoft\Office\<Outlook Version>\Outlook\WebView\Calendar\URL registry key for persistence.

70. **71 Set Outlook Home Page via Registry Let’s hunt it!**

    Search for modification of URL Registry value: (Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13) AND TargetObject:"*\\Outlook\\WebView*" AND TargetObject:"*\\URL" Search for usage of standard Windows tools to create and set the URL Registry value: ((Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1) OR (Channel:Security AND EventID:4688)) AND CommandLine:"*\\Outlook\\Webview*" AND CommandLine:*URL*

71. **72 Outlook Today Page Outlook Today is a handy way**

    to get a quick interactive summary of your calendar, tasks, and messages for the current day. Outlook Today can be used for persistence in the same way as Outlook Homepages. Outlook Today had a menu called data file properties (similar to properties under folders such as Inbox) and through that menu, you could set a homepage value. Unlike Outlook Homepage URL, Outlook Today URL value could not be set remotely and had to be set through the registry under: HKCU\Software\Microsoft\Office\16.0\Outlook\Today\UserDefinedUrl https://medium.com/@b wtech789/outlook-today- homepage-persistence- 33ea9b505943

72. **73 Outlook Today Page example Outlook Today Page allows to**

    execute arbitrary visual basic script, that’s why it is so attractive for adversaries:

73. **74 Outlook Today Page configuration**

74. **75 Outlook Today Page configuration Let’s hunt it! Search for**

    specific Registry values set events: (Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13) AND ((TargetObject:"*\\Outlook\\Today\\Stamp" AND Details:"DWORD (0x00000001)") OR (TargetObject:"*\\Outlook\\Today\\UserDefinedUrl"))

75. **76 Outlook Today Page configuration Let’s hunt it! Search for**

    usage of standard command line tools to set specific Registry values: ((Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1) OR (Channel:Security AND EventID:4688)) AND CommandLine:"*\\Outlook\\Today" AND CommandLine:(*UserDefinedUrl* OR *Stamp*)

76. **77 Outlook Forms T1137.003 – Office Application Startup: Outlook Forms**

    A form is the principal user interface for an item in the Outlook. Outlook provides one or more standard forms for each type of item (mail, contact, and so on). And it is possible to create customized versions of these forms to change the way Outlook displays items. Adversaries may abuse Microsoft Outlook forms to obtain persistence on a compromised system. Custom Outlook forms can be created that will execute arbitrary VBScript code when a specifically crafted email is sent by an adversary utilizing the same custom Outlook form. Once malicious forms have been added to the user's mailbox, they will be loaded when Outlook is started. Malicious forms will execute when an adversary sends a specifically crafted email to the user and user opens it. Unlike rules, there is not an easy way in the UI for a user to check their forms, and also forms can not be seen in OWA, unlike rules.

77. **78 Uses Ruler to create malicious Outlook Forms VBScript code**

    of the malicious form, created by Ruler

78. **79 Outlook Forms. File system artifacts Forms are automatically cached**

    on the client-side in %localappdata%\Microsoft\Forms. This can be used to detect the appearance of new Outlook Forms.

79. **80 Outlook Forms. File system artifacts Let's hunt it! Search**

    for creation of files and folders in %localappdata%\Microsoft\Forms (forms caching repository): Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:11 AND Image:"\\outlook.exe" AND TargetFilename:"*\\appdata\\local\\microsoft\\FORMS*"

80. **81 Outlook Forms. Registry artifacts There are also some specific**

    registry entries created when forms are cached

81. **82 Outlook Forms. Registry artifacts**

82. **83 Outlook Forms. Registry artifacts Let's hunt it! Search for**

    specific Registry values set events: Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:13 AND Image:"\\outlook.exe" AND TargetObject:("*\\BaseMsgCls\\(Default)" OR "*\\FormStg\\(Default)" OR "*\\MsgCls\\(Default)")

83. **84 Spawning suspicious child by Outlook Spawning suspicious child processes**

    (for example, cmd, PowerShell, wscript, cscript) by Outlook is a general anomaly, that can be used for detection of different techniques related to the Outlook features abusing

84. **Spawning suspicious child by the Outlook Let's hunt it! 85**

    Channel:"Microsoft-Windows-Sysmon/Operational" AND EventID:1 AND ParentImage:"\\outlook.exe" AND (CommandLine:(cmd.exe OR "*cmd *" OR *comspec* OR *wscript* OR *cscript* OR *SyncAppvPublishingServer* OR *powershell* *pwsh*) OR OriginalFileName:("cmd.exe" OR "wscript.exe" OR "cscript.exe" OR "SyncAppvPublishingServer.exe" OR "PowerShell.EXE") OR Image:"\\cmd.exe") Channel:Security AND EventID:4688 AND ParentProcessName:"\\outlook.exe" AND (CommandLine:(cmd.exe OR "*cmd *" OR *comspec* OR *wscript* OR *cscript* OR *SyncAppvPublishingServer* OR *powershell* *pwsh*) OR Image:"\\cmd.exe")

85. **86 Questions?**