KAPTOXA Point-of-Sale Compromise

Jan. 14, 2014

iSIGHTPARTNERS℠
Security beyond the edge℠

Contents

Executive Summary

Since Dec. 18, 2013, the U.S. Secret Service (USSS), US-CERT and the cyber threat intelligence firm iSIGHT Partners have been working together to characterize newly identified malware associated with the KAPTOXA operation, which is behind a large-scale point-of-sale (POS) cyber crime breach. This characterization included determining malware functionality and scope, reverse engineering and proprietary research and analysis of threat marketplace activity before, during and after the breach.

The purpose of this release is to provide relevant and actionable technical indicators enabling the identification of additional victims. Since the USSS is actively investigating these breaches, the organizations working on the case are unable to disclose the full nature of the threat at this time, including external connection points, potential attribution or other known tactics, techniques and procedures (TTPs). However, threat updates will be released as appropriate and in coordination with the USSS so as to not interfere with active investigations.

USSS, US-CERT and iSIGHT Partners have confirmed that the high-profile KAPTOXA POS compromise involved the use of a new malware variant, dubbed "Trojan.POSRAM," which is designed to extract payment card details from POS systems. Multiple data points strongly suggest that Trojan.POSRAM was derived from another type of POS malware known as "BlackPOS." At the time of discovery and analysis, the malware had a zero percent anti-virus detection rate, which means that fully updated anti-virus engines on fully patched computers could not identify the malware as malicious.

The KAPTOXA operators also leveraged a variety of other tools to penetrate the targeted network, maintain access and exfiltrate stolen data. While some components of the breach operation were technically sophisticated, the operational sophistication of the compromise activity makes this case stand out. The intrusion operators displayed innovation and a high degree of skill in orchestrating the various components of the activity.

Financially motivated cyber criminals around the world have used POS malware at an accelerating pace for several years. Significantly, POS malware that includes memory scraping capabilities has been available in the Russian-language underground for some time. While Eastern Europe has been the focal point for POS malware development and use, cyber criminals in Brazil have used the technique since at least 2009. Globally, this trend will probably continue because malware offers important cost and risk advantages over hardware skimming techniques.

KAPTOXA Operation Uses Trojan.POSRAM in Sophisticated Retail Compromise

USSS, US-CERT and iSIGHT Partners have confirmed the use of a new Trojan, dubbed "Trojan.POSRAM," in a complex attack targeting payment card information and involving multiple other code types. Trojan.POSRAM is POS malware that monitors memory address spaces used by specific programs. Malware users can specify which programs should be monitored; specific versions of Trojan.POSRAM analyzed by USSS, US-CERT and iSIGHT Partners looked for pos.exe, pp.exe, PosW32.exe and epsenginesrv.exe.

Multiple data points strongly suggest that Trojan.POSRAM was derived from another type of POS malware known as "BlackPOS."

- Both malware types contained unique strings in the programming section of the code:
    - KAPTOXA
    - blocklen:

- .memdump
- GOTIT

- The structure of code is the same between both binaries. BinDiff, a commercial binary diffing tool, identified 875 functions as unchanged and 150 as changed. With a total of 1,294 functions in POSRAM, the BinDiff match rate is 79 percent. IDACompare, a free binary diffing utility, posts similar results. In particular, the Exact CRC match statistics are a very strong indicator. This method takes a CRC hash of the ordered base assembler instructions in a routine and hashes them into a 32-bit integer for comparison. Results are below showing how closely related the two samples are to one another.
- Decompiling both routines using HexRays for the MemMap routine reveals a close association:

blackpos

```
memset(&v6, -858993460, 0x120u);
lpAddress = 0;
v8 = 1879048191;
dword_43D38C = 0;
do
{
  VirtualQueryEx(hProcess, lpAddress, &Buffer, 0x1Cu);
  v7 = unknown_libname_39(v2, v1);
  if ( v7 )
  {
    if ( Buffer.RegionSize )
    {
      v11 = (signed int)Buffer.BaseAddress;
      v10 = Buffer.RegionSize + (_DWORD)Buffer.BaseAddress;
      MemLoop(
        (int)Buffer.BaseAddress,
        (int)hProcess,
        hProcess,
        Buffer.BaseAddress,
        Buffer.RegionSize + (_DWORD)Buffer.BaseAddress);
      ++dword_43D38C;
    }
  }
  lpAddress = (char *)lpAddress + Buffer.RegionSize;
}
while ( v7 && (unsigned int)lpAddress < v8 );
v4 = v1;
_RTC_CheckStackVars(&v13, dword_4028D0);
return unknown_libname_39(v5, v4);
}
```

svchosts

```
memset(&v6, -858993460, 0x120u);
lpAddress = 0;
v8 = 1879048191;
dword_4418A8 = 0;
do
{
  VirtualQueryEx(hProcess, lpAddress, &Buffer, 0x1Cu);
  v7 = unknown_libname_14(v2, v1);
  if ( v7 )
  {
    if ( Buffer.RegionSize )
    {
      v11 = (signed int)Buffer.BaseAddress;
      v10 = Buffer.RegionSize + (_DWORD)Buffer.BaseAddress;
      MemLoop(
        (int)Buffer.BaseAddress,
        (int)hProcess,
        hProcess,
        Buffer.BaseAddress,
        Buffer.RegionSize + (_DWORD)Buffer.BaseAddress);
      ++dword_4418A8;
    }
  }
  lpAddress = (char *)lpAddress + Buffer.RegionSize;
}
while ( v7 && (unsigned int)lpAddress < v8 );
v4 = v1;
_RTC_CheckStackVars(&v13, dword_405340);
return unknown_libname_14(v5, v4);
}
```

The malware is configured to "hook" into these payment application programs to monitor the information they process in memory. These programs are responsible for processing authorization data, which includes full magnetic stripe data (track data). When authorization data is processed, the payment application decrypts the transaction on the cash register system or BOH server and stores the authorization data in random access memory (RAM). The data must be decrypted for the authorization to be completed, so hackers are accessing full track data when it is stored in RAM and using the RAM-scraping malware to steal it.

When the malware identifies this information, it saves it to %windir%\system32\winxml.dll. Every seven hours the Trojan checks to see if the local time is between the hours of 10 a.m. and 5 p.m. If so, the Trojan attempts to send winxml.dll over a temporary NetBIOS share to an internal host (dump server) inside the compromised network over TCP port 139, 443 or 80. This step allows the intrusion operators to remotely steal data from POS terminals with no Internet access.

In addition to Trojan.POSRAM, the following types of code were also used:

- **ICMP Listener:** Listens for custom ICMP packets to log dump transfers from a POS scraper to an internal LAN dump server.
- **Shellcode Loader:** Receives raw commands across the network to be loaded and executed on a compromised host. This tactic is innovative and new to eCrime, able to covertly subvert network controls and common forensic tactics to conceal all data transfers and executions that may have been run through such a loader.

- **Data Manager:** Manages data on the dump server.
- **Exfiltrator:** Trojans that communicate with the centralized dump server to pull stolen data from a temporary DLL file, then exfiltrating it out of the network to a remote FTP server (by IP).
- **Hacking Tools:** The intrusion operators used a variety of admin and hacking tools for network discovery, credential compromise, database operations and port forwarding.

While some components of the breach operation were technically sophisticated, it is the operational orchestration of the KAPTOXA compromise activity that is remarkable.

POS Malware and the Cyber Crime Landscape

Widespread, "commercialized" POS malware is increasingly available on underground marketplaces, which we believe may lead to a demand for private and more specialized POS malware. For example, as banking malware became commercialized and highly visible to law enforcement (e.g., Zeus, Citadel and Carberp) we observed an increased demand for private Trojans. A similar phenomenon may result from the increasing popularity of POS malware.

Numerous types of available POS malware are being sold on the underground, which is making this type of malware increasingly available to cyber criminals. Some of the more popular POS malware is listed below:

- **BlackPOS** (aka "Memory Form Grabber"): POS malware that is easily available due to a leaked version of the source code; the original source code was authored by actor "ree[4]" (for more information and attribution, see iSIGHT Partners. "Analysis of 'Dump Memory Grabber' Point-of-Sale Malware," Malware Report #13-25113. April 8, 2013; and "Attribution for Russian Actor 'Ree[4],' Seller of a Credit Card RAM Memory Grabber," Intel-792666. April 11, 2013).
- **Dexter** (v2 called "Stardust"): POS malware that scans victim machines' process memory for credit card track data and exfiltrates it to a remote command and control (C&C) server (see iSIGHT Partners. "Dexter POS Malware," Malware Report #13-24091. Feb. 6, 2013; and "Publicity Surrounding Dexter Malware Will Probably Contribute to Actor Interest in This Malware in the Underground Marketplaces," Intel- 981126. Oct. 30, 2013).
- **vSkimmer** (Virtual Skimmer): POS malware with a widely available cracked builder and panel (see iSIGHT Partners. "Analysis of vSkimmer Point-of-Sale Malware," Malware Report #13-24544. March 14, 2013).

We believe there is a strong market for the development of POS malware, and evidence suggests there is a growing demand that will continue to drive increased prevalence and availability of POS malware.

- The abuse of online freelance IT marketplaces for the development of POS malware is common. In July 2013, on a popular online freelancing website, more than 20 percent of observed advertisements containing the keywords "POS" and "EMV" were malicious or suspicious, illustrating abuse by cyber criminals in outsourcing the development of POS malware (for more information, see iSIGHT Partners. "Freelancing Website Abuse for POS Compromise Development, Particularly EMV Compromise, Likely to Continue Over Next 6-12 Months," Intel-874657. July 8, 2013).
- This same phenomenon was also observed in September 2010. Average bids for observed, outsourced POS malware projects in 2010 spiked from $425-$2,500 in the first half of the year to $6,500 in the latter part of the year (see iSIGHT Partners. "Adversary Actors Using Global Freelancing Marketplaces to Develop Malware for POS Terminals, Likely Indicating Increased Threat to EMV Payment Systems," Intel-281120. Sept. 3, 2010).

- The market for POS malware is further exemplified by multiple cyber criminal advertisements interested in obtaining this type of malware. For example, there was an increase in observed interest in POS malware among French-speaking cyber criminals in late 2013.

We suggest that the spread of POS malware will primarily be enabled by further development of existing credential theft Trojans rather than the creation of entirely new malware families—particularly as there is evidence of this already occurring—although original development is also probable. For example, ProjectHook (RAM-scraping malware) is based on Zeus, and one actor has already claimed to have created a new builder and panel for vSkimmer, most likely based on the alleged leak of the original.

Leaked source code of credential theft malware could provide a starting block for actors who do not have the skill to create an entirely new type of malware from scratch, or for actors seeking to leverage previous work to optimize the efficiency of their scheme. Such lowered barriers to market entry could lead to more types of POS malware offered for sale and therefore eventually lead to cheaper prices and larger user bases (for a general outlook on POS malware, see iSIGHT Partners. "Command and Control Infrastructure Used for Three Types of POS Malware Demonstrates Growing Popularity and Availability of Such Cyber Crime Tools," Intel- 963893. Oct. 15, 2013).

Appendix 1: Initial Recommended Mitigation Strategies

Look for the following generic indicators, which may reveal a compromise:

- Audit networks for possible rogue PING messages that contain custom text messages.
- Audit hosts for a rogue "POSWDS" service.
- Look for rogue applications in memory that may attempt to masquerade as svchost and/or other programs on POS terminals.
- Look for data dumps within temporary DLL files stored on POS terminals, such as %windir%\system32\winxml.dll.
- Look for a rogue data manager application on internal LAN servers.
- Look for unauthorized FTP exfiltration on Internet-accessible hosts/servers.

Mitigation may be very complex and involve the immediate removal of known malware for the architecture of this attack, extensive audits and response work within the entire network, changes to accounts, passwords and other data that may have been compromised internally and coordination with iSIGHT Partners and law enforcement in an active investigation.

Appendix 2: Technical Malware Analysis

The following technical information is derived from malware analysis performed by iSIGHT Partners and is intended to allow those potentially affected by similar activity to check their systems for potentially malicious activity. Network indicators (and specifically, IPs) linked to this attack have been redacted due to ongoing law enforcement investigations.

POSWDS Service Created by Malware

When run, the Trojan creates a service called "POSWDS" and runs the code from the original location of execution.

Trojan monitors memory space for different programs (observed targeted programs include pos.exe, pp.exe, PosW32.exe and epsenginesrv.exe, depending on the variant) to steal sensitive information from memory, incrementally saving data to %windir%\system32\winxml.dll.

The Windows registry is modified to contain or modify keys to configure the service and disable proxy:

HKLM\SYSTEM\ControlSet001\Enum\Root\LEGACY_POSWDS\0000\Control
NewlyCreated = 0x00000000
ActiveService = "POSWDS"

HKLM\SYSTEM\ControlSet001\Enum\Root\LEGACY_POSWDS\0000
Service = "POSWDS"
Legacy = 0x00000001
ConfigFlags = 0x00000000
Class = "LegacyDriver"
ClassGUID = "{8ECC055D-047F-11D1-A537-0000F8753ED1}"
DeviceDesc = "POSWDS"

HKLM\SYSTEM\ControlSet001\Enum\Root\LEGACY_POSWDS
NextInstance = 0x00000001

HKLM\SYSTEM\ControlSet001\Services\POSWDS\Enum
0 = "Root\LEGACY_POSWDS\0000"
Count = 0x00000001
NextInstance = 0x00000001

HKLM\SYSTEM\ControlSet001\Services\POSWDS\Security
Security = 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80 14 00 FF 01 0F 00 01 01 00 00 00 00 00 01 00 00 00 00 02 00 60 00 04 00 00 00 00 00 14 00 FD 01 02 00 01 01 00 00 00 00 00 05 12 00 00 00 00 00 18 00 FF 01 0F 0

HKLM\SYSTEM\ControlSet001\Services\POSWDS
Type = 0x00000110
Start = 0x00000002
ErrorControl = 0x00000000
ImagePath = "file and pathname of the sample #1"
DisplayName = "POSWDS"
ObjectName = "LocalSystem"

FailureActions = FF FF FF FF 01 00 00 00 01 00 00 00 03 00 00 00 74 00 6D 00 01 00 00 00 A0 86 01 00 01 00 00 00 A0 86 01 00 01 00 00 00 A0 86 01 00

HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_POSWDS\0000\Control
NewlyCreated = 0x00000000
ActiveService = "POSWDS"

HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_POSWDS\0000
Service = "POSWDS"
Legacy = 0x00000001
ConfigFlags = 0x00000000
Class = "LegacyDriver"
ClassGUID = "{8ECC055D-047F-11D1-A537-0000F8753ED1}"
DeviceDesc = "POSWDS"

HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_POSWDS
NextInstance = 0x00000001

HKLM\SYSTEM\CurrentControlSet\Services\POSWDS\Enum
0 = "Root\LEGACY_POSWDS\0000"
Count = 0x00000001
NextInstance = 0x00000001

HKLM\SYSTEM\CurrentControlSet\Services\POSWDS\Security
Security = 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80 14 00 FF 01 0F 00 01 01 00 00 00 00 00 01 00 00 00 00 02 00 60 00 04 00 00 00 00 00 14 00 FD 01 02 00 01 01 00 00 00 00 00 05 12 00 00 00 00 00 18 00 FF 01 0F 0

HKLM\SYSTEM\CurrentControlSet\Services\POSWDS
Type = 0x00000110
Start = 0x00000002
ErrorControl = 0x00000000
ImagePath = "file and pathname of the sample #1"
DisplayName = "POSWDS"
ObjectName = "LocalSystem"
FailureActions = FF FF FF FF 01 00 00 00 01 00 00 00 03 00 00 00 74 00 6D 00 01 00 00 00 A0 86 01 00 01 00 00 00 A0 86 01 00 01 00 00 00 A0 86 01 00

HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Internet Settings ProxyEnable = 0x00000000

HKLM\SYSTEM\ControlSet001\Control\ServiceCurrent
(Default) =

HKLM\SYSTEM\CurrentControlSet\Control\ServiceCurrent
(Default) =

Every seven hours the Trojan checks to see if the local time is between the hours of 10 a.m. and 5 p.m. If the local time is during the time range specified, the Trojan attempts to exfiltrate winxml.dll over a temporary NetBIOS share to a host on the internal network. This most likely occurs over TCP port 139; however, NetBIOS can also fall back upon the WebDav transport, which uses ports 443 or 80. In a

common networked environment the Trojan can easily communicate via NetBIOS over the network using the aforementioned ports.

Three commands are used to move data from a collections host to the internal LAN dump server. The commands are used to mount a drive, move data to the remote host, and then the mapped network share is removed as a way to conceal communications.

ICMP Listener

Several executables in this incident are designed to listen for ICMP (ping) messages across the LAN, with embedded status updates about dumps transferred to the internal dump server. This is done as a way to log dumps sent to a dump server, covertly across the LAN, prior to exfiltration.

A POS scraper transfers stolen data to an internal dump server. It sends a status update (via an embedded string with an ICMP packet) across the network, which is then picked up by an ICMP listener, which logs the event to a file at the file log.txt in the applications home directory and displays the text message to a console window. Early analysis strongly suggests that this specific sample was likely used as a way to test functionality on an internal platform server and ICMP logging of dumps, prior to rolling out an attack on another internal LAN dump server seen in this attack.

Data Manager

A data management Trojan that runs on a compromised internal C&C server, this program is managed by remote Exfiltrator code. It is responsible for copying all stolen log data to a temporary storage file. All log files found within the folder c:\windows\twain_32\ will be copied into the file c:\windows\twain_32a.dll. As seen with POS scraper Trojans in this attack, the DLL is only a temporary storage file for stolen data, and the file is deleted once a transfer has been completed.

Exfiltrator

Similar to POS Terminal Trojans in this attack, Exfiltrator Trojans communicate with a C&C that receives aggregate stolen data, ready for exfiltration:

Each Exfiltrator is designed to send stolen log data to a remote computer.

Functionality for the code is as follows:

COMMAND: ftp -s: [Application Path\\cmd.txt

- start data management utility XXX on remote dump server to conglomerate log files
- sleep for four minutes
- killing the executable again
- copying over stolen data from dump server
- generate an FTP upload script to upload to *omitted_IP*\public_html\cgi-bin
- execute the FTP script (Windows FTP client) for exfiltration

This sample PULLs data from the internal drop server for FTP exfiltration.

Shellcode Loader

Shellcode Loader binaries can easily appear to be legitimate tools due to strings included within the binaries to trick responders and forensic experts. They are all designed to download second-stage shellcode and execute it, covertly, without leaving tracks/logs behind of what was run on a host.

Network traffic of an executable code being transferred to the Shellcode Loader would just look like a binary blob with high entropy. There is no NOP slide to trigger shellcode detections, no MZ header of an executable and probably no strings because it is traditional to encode shellcode and prefix it with its own encoder. Additionally, no files are necessary for the loaders to run code. In addition, this technique leaves no traces in memory, making it very difficult to identify what might have been transferred to and run on the compromised host.

These loader applications include the publicly available Harmony API hasher written by Stephen Fewer. The specific application of this technique for running shellcode appears to be innovative and unique to the architecture of this attack, for covert operations.

Various Hacking Tools

There are a significant number of various hacking tools used in this attack for network discovery, credential compromise, database operations and port forwarding. Specific details on these files have been omitted due to the ongoing law enforcement investigations.

File Details

Note: A multi-scanner of all samples at the time of analysis revealed a zero percent detection (undetected, formerly unknown family of code). Various hacking tools are generally detected at various rates, as they are potentially unwanted programs in most instances. It appears likely that codes for this attack were customized to avoid detection and to communicate to an internal LAN dump server for exfiltration, as demanded by the network architecture.

Name: ab6fb405ef8f06ee98be0b9da5250607
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 245760
Md5sum: ab6fb405ef8f06ee98be0b9da5250607
Sha1: 03d34733b77cf35ada4568d557b2a41e063ad6ee
Sha256: 59a7a979da859d625cf061bb5626efe465a253f196fcfb8338a087bda308bd0b
Fuzzy: 6144:xRZLUmGTVeHY/w4lCl38dmlJil5qfrwwYE46YEA:xRQgww48omlJifBwYE46

Name: 93405c57e915680f0182650fb75c47ee
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 737280
Md5sum: 93405c57e915680f0182650fb75c47ee
Sha1: f590db3cca3a3c51bcd41b48237104a39df27976
Sha256: a70656d40a64170bcae021e989fc08bbaed608a6c437979dfec3171e71c9e9b8

Fuzzy:
3072:Xa9cqx3GQCL911QnEk4YaoJMMnAgyBQv4oWsIZzoMWWLhGUaWj9qD8xeU7i5K:XaJxWQCL91KEV
hgyGgofEpWkMdWj40

Name: 3f00dd56b1dc9d9910a554023e868dac
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 94208
Md5sum: 3f00dd56b1dc9d9910a554023e868dac
Sha1: b6b55455f08f46f972133de6cb94498ccba8b035
Sha256: 436d23a55ad776297439871e4b05af7467d243e039b07331b505ec2a71bc884a
Fuzzy: 1536:nr+GT4HQqoLEdIcTKcTMDVmBXd9pmOjbo4m/:rjVLUIcTiuoOjboT/

Name: 433a2750429d805907aa4848ff666163
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 90112
Md5sum: 433a2750429d805907aa4848ff666163
Sha1: 535371095921bf319911feb54bf9d9bee57f8f9a
Sha256: 92c83543242c3db1e6cfa8e8f7977dbddb9d5604341a00458707e775567237f5
Fuzzy:
1536:yx0xAtdC4tTI96JnSNiRXqzy1kYX/9cB1BE2ODFtnk5DFojWsflx:q0ibC4tTI9qnSGXVkU/9A1a2ODFtnk5
I

Name: 65dd8d2d9604d43a0ebd105024f09264
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 421888
Md5sum: 65dd8d2d9604d43a0ebd105024f09264
Sha1: ab354242992af39f93520ac356ec12796e119151
Sha256: 6affbc089af37728beab3a27756f5eac470a366e29cfb6d2a58953fee3124b61
Fuzzy: 12288:VkWMVrsccmm6PwvJdDndszu5SXIRoleWRb/6:VJJ0zuEXnB/

Name: 4352e635046aa624dff59084d5619e82
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 315392
Md5sum: 4352e635046aa624dff59084d5619e82
Sha1: 9926d7446a9311e2b36f45c1759cd38e5e25f5bf
Sha256: 34c954a988e66345358f8e1accf7ad16d13a49496b84e239dd3656f6612d5a58
Fuzzy: 6144:50BXBKytgz0EWNvbv8s3K5aEMmNeZ/pqZ0gFRxAyndrJnhO:aKytgz0E6z3DtZ4Tc2hO

Name: 0b33b4d61ea345f16c4a34b33e9276bc
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 102400

Md5sum: 0b33b4d61ea345f16c4a34b33e9276bc
Sha1: 8b318d4c525c31159ba3ade7cd4192179c8c85be
Sha256: e687798efb89213f7e7cff916a4a265e26d2af9d9703e70e82683d1de0f96398
Fuzzy:
768:burrAUdQwSzRmLqtJiWXt91lcYQpMEbkt9IIv76S83wKvhnwYt6Pha6j:burrhyw7qDiWXfQe4kteS83d1
xYPrj

Name: 6c1bcf0b1297689c8c4c12cc70996a75
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 111104
Md5sum: 6c1bcf0b1297689c8c4c12cc70996a75
Sha1: 9d99a2446aa54f00af0b049f54afa52617a6a473
Sha256: 40dc213fe4551740e12cac575a9880753a9dacd510533f31bd7f635e743a7605
Fuzzy: 3072:xRrDKrIdBh3D3GA20Cqx/V8pt4TQtnoWB+:xAsnhrGAzCqLEt48n

Name: 453810a77057d30f0ee7014978cdc404
Identifier: attacker
Extension: zip
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 319488
Md5sum: 453810a77057d30f0ee7014978cdc404
Sha1: 39f9f9db004da35fd58f5a4ca937a584f7492050
Sha256: 7976a84f89a27b3e73b30580cb55842c9aba7476b18f842db55d8c4fb1b42357
Fuzzy: 6144:MAuKxxFZFfBrBrQdpVMRk+ELKP+p/o+7mqweqkJbIYaz+:JuKfFZFfBVyC7E0wh7keGYy+

Name: 08644155f5c8f94f0cc23942c5c5068f
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 311296
Md5sum: 08644155f5c8f94f0cc23942c5c5068f
Sha1: dd57533c9deb80d1b10a75300fc11cf8fe779f19
Sha256: af5cf9f9b9418885b1027ca8c8bca34ebe7c628ef838d50ce7ee18f7632718db
Fuzzy: 6144:YQpbqTJNTiOKb7IN9opX9XHxOaygkoA5G+JunADuAe:NpbqTJNTiOcdZRXyhoA5pICe

Name: 623e4626d269324da62c0552289ae61f
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 360448
Md5sum: 623e4626d269324da62c0552289ae61f
Sha1: 03fb0385e6d6f1c5613f38af743d057e770a0244
Sha256: c856e226ab8292b6d5827a03120ce6f629c77f9196b71dac0965bf47e747b438
Fuzzy:
6144:7OZseaZoYVw4GhHLVOWTYBLanhuk5eZM5pzF8nd9MHjoP:vjeYVw4GhHgQyLYhLeZM5wd9MI

Name: 290c26433a0d9d14f1252e46b1204643
Identifier: attacker
Extension: exe

Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 360448
Md5sum: 290c26433a0d9d14f1252e46b1204643
Sha1: 379ba2d30ada59ca7fba71c594840f3caec86d4f
Sha256: e67d435134de9a113986d40b1b053e0134c79328859c95abb845692c2c8487cd
Fuzzy:
6144:Z/ZKO6ZJnw80cHUayppyBoKehuk5XXespiyl8pd9OHXAP:Czvnw80cHefAodhLXXes0tT9Os

Name: e2db09553f23a8abc85633f6bf1a0b49
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 249856
Md5sum: e2db09553f23a8abc85633f6bf1a0b49
Sha1: ac69c8a62c7d306ac56c8cdf6d738fa8115f1600
Sha256: 5c8b6a629c77bbed2e1ee78c46d9df550ddebfa511be92864e0895cc7cc0f832
Fuzzy: 6144:OdYqcN0GJeDDzo2M4qo5BHetNLljmoNbUjJf:OdIEg2FpB+tNLlRbUjJ

Name: 322e136cb50db03e0d63eb2071da1ba7
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 286720
Md5sum: 322e136cb50db03e0d63eb2071da1ba7
Sha1: 332548d0bc638c8948f3a429e79053003b4f6261
Sha256: 242c4bb74dc6962d9ebb52fa8dbfd8cd5173423aafe9b65204c39cc43a810722
Fuzzy: 6144:Zs1TEC9tjlimXZ3dX3iIMWHbn5rkfFEAKGlLIT0s9L:O1TEC9tjlimJ3l175rkc0s9L

Name: 322e136cb50db03e0d63eb2071da1ba7
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 286720
Md5sum: 322e136cb50db03e0d63eb2071da1ba7
Sha1: 332548d0bc638c8948f3a429e79053003b4f6261
Sha256: 242c4bb74dc6962d9ebb52fa8dbfd8cd5173423aafe9b65204c39cc43a810722
Fuzzy: 6144:Zs1TEC9tjlimXZ3dX3iIMWHbn5rkfFEAKGlLIT0s9L:O1TEC9tjlimJ3l175rkc0s9L

Name: a35e944762f82aae556da453dcba20d1
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 359936
Md5sum: a35e944762f82aae556da453dcba20d1
Sha1: a7c8b8abd907a73752ce5476e567ddac1b794b8f
Sha256: 55fa6b579f7a3f06ad3b28d458e42462a392be7b116b762ff7b9f659138d35e8
Fuzzy:
6144:MEZS9aZUZwdhlwEblU7Qw3+r19hu0PWdp9l0HeNB3U5w:+8SZw/lwEC8Vr/h9PWdi+NBT

Name: f4bdc5e507d887d5d2cd2c4c61cfcfe1
Identifier: attacker

Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 86128
Md5sum: f4bdc5e507d887d5d2cd2c4c61cfcfe1
Sha1: a737c709d5f61d1b0e4b9822cbf704e96736fac6
Sha256: 85d39c64b88592887e4c4ef0b0faeccee7c8ce60d8cde7cd82d62b5571f6296e
Fuzzy:
1536:WbQhb3eueL8yGvLzth6NvS6pBChl7uxJ/3VKbY+RONEBo55S4iGjotB:WUFeonFheS65r/eYoOyBo55SH2s

Name: 02137a937f6fbc66dbc59ab73f7b1d3e
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 53299
Md5sum: 02137a937f6fbc66dbc59ab73f7b1d3e
Sha1: 2f2c6beba902d95486e01608e58ecde9ee7a7bfc
Sha256: ec19350d31d78d2ae04ca3c0741e4ccf16effeb44ed957b1faf3719376ce0b3f
Fuzzy:
768:VxgDAUZFV9WnLlA3X593ElTEt05kugg7jmnoZEH9My5ujnPZnN8R3Dk9YGZsX6r:VY/+neT09u05kuggHmnoZioD8Rysqr

Name: 4b9b36800db395d8a95f331c4608e947
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 122880
Md5sum: 4b9b36800db395d8a95f331c4608e947
Sha1: 608a63f8a6d981196303164bd0962336aa6a86c5
Sha256: 777068fee7af698a7e1445547285d7525d5865c06489cd7839596d761b075246
Fuzzy: 3072:ekrLWJoNO5MEn9KWjVg6djMk07NYXYernzga0F:eALWPMbUKojV0pYXY2

Name: df5dbcbcac6e6d12329f1bc8a5c4c0e9
Identifier: attacker
Extension: rll
Type: PE32 executable for MS Windows (DLL) (console) Intel 80386 32-bit
Size: 17432
Md5sum: df5dbcbcac6e6d12329f1bc8a5c4c0e9
Sha1: 2b18897cc597b9c6be1abbc9688fb154313541b1
Sha256: 1a57eee6cdbb31b564ab75ef0d0417e7d48fb796de93777388682e76e9c252c3
Fuzzy:
192:jT8PWYmW/9HDh4vMYtZ2WVHZso6oEQKPnEt2yt8mJz+Hz+ehjT4NmVR:P8PWYmWNDmMY7B1nELKt8Cu1j0

Name: 814b88ca4ef695fea3faf11912a1c807
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 53248
Md5sum: 814b88ca4ef695fea3faf11912a1c807
Sha1: 3cbf7a6ab29172d78b63f68d814359b72cda6057

Sha256: 37175f167f355da8d69cd597c60c70d7d6f9d154d8578d68fdcb43cb20ca55d8
Fuzzy: 768:KQAun71rIjCnyGRwZ1ZateO3a4Zgb2fH72Vld:KM1roCLOZ1eeXT2Ald

Name: d975fc6cda111c9eb560254d5eedbe0a
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Size: 45056
Md5sum: d975fc6cda111c9eb560254d5eedbe0a
Sha1: e8fcb3b02240ca7a67fc9e67245bdfb1d0ccd14f
Sha256: 674709fa4a0ad41f675a799d41429b9f78fe6d51dd6a97d539ee01e37d1e9148
Fuzzy: 768:0p1mxlgrSX/Z5Cx9XoWYAmx8shUQrkGOMj/:0p1MrSvax9tYAmx8sh9hOMj

Name:
Identifier: attacker
Extension: msi
Type: CDF V2 Document, Little Endian, Os: Windows, Version 5.2, Code page: 1252, Title: Installation Database, Subject: Audit security policies, examine network security and recover account passwords, Author: Elcomsoft Co. Ltd., Keywords: password, password recovery, lost password, recover password, remove password, remove protection, recover account, unlock password, reset password, forensics software, system software, security software, ElcomSoft Password Recovery Bundle, forgot administrator password, forgot windows password, vista password, distributed password recovery, nVidia, GPU, archive, ZIP, RAR, ARJ, Microsoft Word, Microsoft Excel, Microsoft Access, Microsoft Outlook, Microsoft Project, Microsoft PowerPoint, Microsoft OneNote, Microsoft Money, Microsoft Visio, Microsoft Publisher, VBA, Visual Basic for Applications, backdoor, attack, rainbow tables, thunder tables, bruteforce, Adobe Reader, PDF, database password, Microsoft SQL Server, Microsoft SQL Server Express, MSSQL, MS SQL, Corel WordPerfect Office, WordPerfect, Quattro Pro, Paradox, Lotus Organizer, Lotus WordPro, Lotus 1-2-3, Lotus Approach, Freelance Graphics, Intuit Quicken, Quicken Lawyer, QuickBooks, ACT! software, ACT, Symantec, Best Software, Sage, Microsoft Internet, Comments: ElcomSoft Password Recovery Installer, Template: Intel;1033, Revision Number: {17EC52E6-8FBE-415E-B233-4D5CF02288E8}, Create Time/Date: Thu Aug 15 07:47:32 2013, Last Saved Time/Date: Thu Aug 15 07:47:32 2013, Number of Pages: 200, Number of Words: 2, Name of Creating Application: Windows Installer XML (3.0.5419.0), Security: 2
Size: 9669632
Md5sum: 793860864d74ee6ed719d57b0a3f3294
Sha1: 4162e07aed718d8437457134ab6527999d4a4437
Sha256: f5610a46496d42b12e257c7326dd5bc79ff56ead8229772396c24a1ca2a4d297
Fuzzy: 196608:db8+Jq6j6rDa4pY8DXHNAD3sL3dIPZlGZcbCvQvr9:dt1Ma4pYid9ZcP

Name: aeee996fd3484f28e5cd85fe26b6bdcd
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 381816
Md5sum: aeee996fd3484f28e5cd85fe26b6bdcd
Sha1: cd23b7c9e0edef184930bc8e0ca2264f0608bcb3
Sha256: f8dbabdfa03068130c277ce49c60e35c029ff29d9e3c74c362521f3fb02670d5
Fuzzy:
6144:xytTHoerLyksdxFPSWaNJaS1I1f4ogQs/LT7Z2Swc0IZCYA+l82:x6TH9F8bPSHDogQsTJJJK+l82

Name: 2cd8dddaf1a821eeff45649053672281

Identifier: attacker
Extension: zip
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 1007616
Md5sum: 2cd8dddaf1a821eeff45649053672281
Sha1: aa18f3efc7ff2af88e63db7833c3b2a58a8a7748
Sha256: cdf65f15a5bb26341f090f9a07aa4dc8eede5e314885d547757bcc5e87f2deb6
Fuzzy:
6144:tq2y0CwKsPGXWLsj+YcBx9WKRmM4oXBMfWx891P94RF3/PoLx:02y0WEtLK+V4oWFP94R5/Pot

Name: a109c617ecc92c27e9dab972c8964cb4
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit Mono/.Net assembly
Size: 126976
Md5sum: a109c617ecc92c27e9dab972c8964cb4
Sha1: 304b4ae488d87449f11a2cae4f5d1eb6def8b104
Sha256: e25e75196fecf1991fdb1d7db4413662e9189ee5f3d8b91dd11e58a7aec2a38a
Fuzzy:
1536:3Bd/UgCokjhSYwQz8QeUhRnHcwV3atrossRLCzmsg8cxg+1GnNZ+WhVPkQV/dVUI:Rd/UtpVWsRLM
msg8cXC8I/3UI

Name: f6877447d2bd0199ad2f073a391aacde
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 251904
Md5sum: f6877447d2bd0199ad2f073a391aacde
Sha1: fb601f94cc0ef6648b3056c2826d8821c594a860
Sha256: c9ca6ed8beb91b863f7dee8bd44bd46af32672ae5361b586765ada8aaeb6e8e2
Fuzzy: 6144:Td9V/ZZUXZ4g5NLO4thzIJWjP3ukvYTdABg:Tx/ZZKZF5NLO47MJkPfgTdUg

Name: e2db09553f23a8abc85633f6bf1a0b49
Identifier: attacker
Extension: exe
Type: PE32 executable for MS Windows (console) Intel 80386 32-bit
Size: 249856
Md5sum: e2db09553f23a8abc85633f6bf1a0b49
Sha1: ac69c8a62c7d306ac56c8cdf6d738fa8115f1600
Sha256: 5c8b6a629c77bbed2e1ee78c46d9df550ddebfa511be92864e0895cc7cc0f832
Fuzzy: 6144:OdYqcN0GJeDDzo2M4qo5BHetNLljmoNbUjJf:OdIEg2FpB+tNLlRbUjJ
Anti-Virus Scan Results