Threat Intelligence Gathering, Malware Collection and Incident Response Proposal

"Discover, Investigate and Report"

United States Military Academy & University of Detroit Mercy

Independent Study Class for Fall 2006

Supervisor:
Director, Information Technology and Operations Center, Lieutenant Colonel, Ronald C. Dodge JR., Ph.D.

By:
Moe Shajera

Purpose of the project:

The purpose of this project is to establish a unique and complete process for threat intelligence gathering malware collection and incident response. This project will address how members of the honeynet alliance can contribute to the process of gathering threat intelligence and collecting malware.

Members of the honeynet alliance can participate and contribute malware collected with the help of nepenthes or any suspicious scripts being seeded in the honeynet they are monitoring. Access will be granted to all the collected malware and will be waiting for special process escalation to determine the uniqueness of the malware and if so, the appropriate IT security incident response government agencies and other organizations will be notified.

Honeynet Threat Intelligence Center (HoneyTIC):

The Honeynet Threat Intelligence Center (HoneyTIC) will be introduced to manage the threat intelligence gathering malware collection and the incident response effectively.

This center will be responsible for gathering threat intelligence such as new vulnerability, exploited or a Malware in the wild spreading through a newly discovered vulnerability.

HoneyTIC will be responsible for analyzing, providing accurate prioritization of the threat being looked at and prioritizing the importance of the malware being captured and analyzed based on the threat intelligence findings. To make this process easy, it will be divided into three phases:

1. Discovering Phase
2. Investigating phase
3. Reporting Phase

Discovering Phase:

To make the discovering phase much easier, it will be divided into two critical processes:

1. Threat Intelligence Gathering Process:

This process will be established so members of the honeynet alliance can easily monitor threat intelligence sources throughout the world for known and new security events and unknown activities. This process will give the HoneyTIC a clear picture of new and emerging threats as they arise.

2. Automating Malware Collection Process:

Automate the complete process of capturing a malware binary, analyzing it, and finally determining whether or not the malware has been discovered already.

For collecting and analyzing malware, various open source tools that have been released by the members of the honeynet alliance or other IT security organizations that are interested in this subject can be used.

In order to make the process of analyzing malware faster, the automation Malware Collection Process needs to be come up with. This process starts with finding a malware collector for tools and malware feeder source(s) to provide us with the malware that needs to be looked at and analyzed in more detail.

By using the MWCollect/Nepenthes tool, it is simple to collect the malware which is local to any owned network. mwcollect is an easy solution to collecting worms and other autonomous spreading malware in a non-native environment like FreeBSD or Linux.

When combined with the "automated Norman Sandbox analysis reports", this represents an excellent way of keeping on top of malware activity without the need to deploy full high interaction honeypots.

A free tool released by LURHQ Corporation is also being researched here. This tool is called (Truman - The Reusable Unknown Malware Analysis Net).

Truman can be used to build a "sandnet", a tool for analyzing malware in an environment that is isolated, yet provides a virtual Internet for the malware to interact with. It runs on native hardware, therefore it is not stymied by malware which can detect VMWare and other VMs. The major stumbling block to not using VMs is the difficulty involved with repeatedly imaging machines for re-use. Truman automates this process, leaving the researcher with only minimal work to do in order to get an initial analysis of a piece of malware. URL Source http://www.lurhq.com/truman

Investigating phase:
in this phase, in-depth investigation will be done if one of the following circumstances occurs:

- New vulnerability being released but not confirmed
- If we don't have a sample being captured by our Malware collection tools, we need to investigate more and acquire a sample.
- Questionable scripts sent by a member of our honeynet alliance.
- ...etc.

After determining that the captured malware is unique such as it is believed to exploit a new vulnerability, analyst need to follow a process called the "Escalation Process" ' more on this will be covered throughout the research; the process is being developed.

Reporting Phase:
If the malware that has been captured is believed to exploit a very urgent un-patched vulnerability, a **"Reporting Process"** will take effect and the appropriate people will be notified.

It will also be a good idea to deliver a Daily, Weekly, and Monthly Honeynet Alliance Intelligence Gathering Report Summary describing the findings and any malware trends that are being seen.

Threat Intelligence Gathering, Malware Collection, Malware Analyses, Incident Response and Reporting

"Discover, Investigate and Report"

United States Military Academy & University of Detroit Mercy

Independent Study for Fall 2006

Supervisor:
Director, Information Technology and Operations Center, Lieutenant Colonel, Ronald C. Dodge JR., Ph.D.

By:

Moe Shajera

Introduction:

Malware and spam perforation has been a global threat to the Security community ever since the existence of the Internet. The use of sophisticated and centralized command and controlled botnets throughout the Internet has helped the most with the spread of malware and spam.

Because of this autonomous spread of malware and the nonstop development of new threats and attack techniques, there exists a need to deploy new or improved security tools and develop new techniques and processes to help the security community mitigate threats.

The purpose of this research is to establish a unique and complete process for threat intelligence gathering, malware collection, incident response and reporting. The prototype of this system will be deployed in the Information Technology and Operations Center (ITOC) at the United States Military Academy (USMA). The ITOC is an information technology and security research center with an active effort in malware detection. Through its involvement with the honeynet alliance, the ITOC is well suited to leverage the expertise of other security professionals.

This research will also address how members of the malware collection alliance can contribute to the success of this whole process. This research is also meant to establish a way to strengthen the communication and sharing of sensitive and critical information. Additionally, we intend to show how certain security tools that have been developed by the malware collection alliance and other independent security tools can help the malware analysts achieve the above objectives.

What is the Malware collection alliance?

According to the mwcollect alliance, the point is to create a trusted community which aims at collecting malware. All participants contribute data (e.g., malware collected with the help of nepenthes) then have access to all data contributed by others. The central repository is now up and running and the mwcollect alliance has about 100 participants.

In the following sections, we will find out what should be done to establish the above mentioned processes:

Honeynet Threat Intelligence Center (HoneyTIC):

In order to link the malware analysts together, there is a need for a command center. The Honeynet Threat Intelligence Center (HoneyTIC) will be established to manage the threat intelligence gathering, malware collection, incident responses and reporting more efficiently.

HoneyTIC needs to be a very secure web-application. The data uploaded to the HoneyTIC may contain 0-day exploits that are not in general distribution. Access to the secure area will be granted to members of the malware collection alliance only. This center will be the main central intelligence place for the malware analysts and will be responsible for gathering threat intelligence such as security breaking news, new vulnerabilities, exploits or newly discovered malware.

The HoneyTIC will be responsible for analyzing current threats, providing accurate prioritization of the threat being looked at and prioritizing the importance of the malware being captured and analyzed based on the threat intelligence findings.

The HoneyTIC is considered to be the central intelligence gathering for everything related to the spread of malware. To accomplish this, the HoneyTIC intelligence gathering process needs to be divided into three phases:

1. Discovering Phase
2. Investigating phase
3. Reporting Phase

1st Phase (Discovering):
It is very important for malware analysts to discover new threats in their very early stages. Early threats or attack discoveries allow incident security respondents to take the proper steps to mitigate security risks. Discovering security threats and exposures can be accomplished by analyzing historical trends, monitoring real-time security events and identifying security threats against normal activities.

To make the discovering phase much easier, it is divided into two critical processes:

1. Threat Intelligence Gathering
2. Malware Collection

Threat Intelligence Gathering:
It is imperative to first identify what type of threat intelligence security analysts are trying to gather. Threat intelligence gathering is needed to present malware analysts with the most current threat and attack context.

This threat intelligence gathering will provide highly detailed information such as the latest security breaking news, security incidents and newly un-

patched vulnerabilities for all security analysts. This is to help with identifying the scale of the threat and leading them to reporting these findings to the appropriate security organizations.

The gathered information will also help malware analysts assess current security threats, detect or identify new types of attacks and respond to incidents. Threat intelligence gathering will enable security analysts to discover unknown threats and respond faster to security incidents.

A dedicated section on the HoneyTIC will be for gathering and viewing the latest threat intelligence. This Web-based application will present malware analysts with the latest gathered security information from multiple sources throughout the internet and present this security data via a single interface. This will give security analysts a holistic understanding of the Internet security status. This service will also present security analysts with the information needed to determine the right courses of action and accurately prioritize tasks.

Being able to provide all the latest global security threats in one place will enable the security analysts to keep an eye out on any global threats in its early stages such as any possible malicious attacks. Lastly, the threat intelligence gathering process will result in publishing the daily intelligence gathering summary.

To accomplish the above objectives, the malware analysts must keep an eye out for updates on security threats. This can be accomplished by checking security dedicated sections such as Security Weblogs, Security Response Centers and the Latest Virus Threats sections which are found on AV vendors, government security incident response agencies, and other private specialized security organization Web sites.

Rigorous research has been done to identify the top and most important security Web sites for threat intelligence gathering. The following are devoted security organizations that can help with this part:

Security WebLogs:

1. http://www.f-secure.com/weblog
2. http://isc.sans.org/diary.php
3. http://www.viruslist.com/en/weblog
4. http://www.websensesecuritylabs.com/blog
5. http://sunbeltblog.blogspot.com/

Internet Traffic and Threat Analysis:

1. http://www.securitywizardry.com/radar.htm

2. http://isc.sans.org/index.php
3. http://www.dshield.org
4. http://www.internetpulse.com
5. http://www.internettrafficreport.com
6. https://analyzer.securityfocus.com
7. http://www.legions.org
8. http://www.advisories.nl

Latest AV Vendors Malicious Code Threats:

1. http://www.symantec.com/avcenter/defs.added.html
2. http://www.symantec.com/enterprise/security_response/threatexplorer/threats.jsp
3. http://www.trendmicro.com/vinfo
4. http://de.trendmicro-europe.com
5. http://vil.nai.com/vil/newly-discovered-viruses.asp
6. http://vil.nai.com/vil/content/alert.htm
7. http://www.f-secure.com/v-descs/_new.shtml
8. http://www.viruslist.com/en/viruses/alerts
9. http://www3.ca.com/securityadvisor/virusinfo/default.aspx
10. http://www.sophos.com/downloads/ide
11. http://www.fortinet.com/FortiProtectCenter

Malicious Code Prevalence Sites:

1. McAfee
2. Message Labs
3. Symantec
4. Trend Micro
5. ISS
6. Norman
7. Internet Storm Center

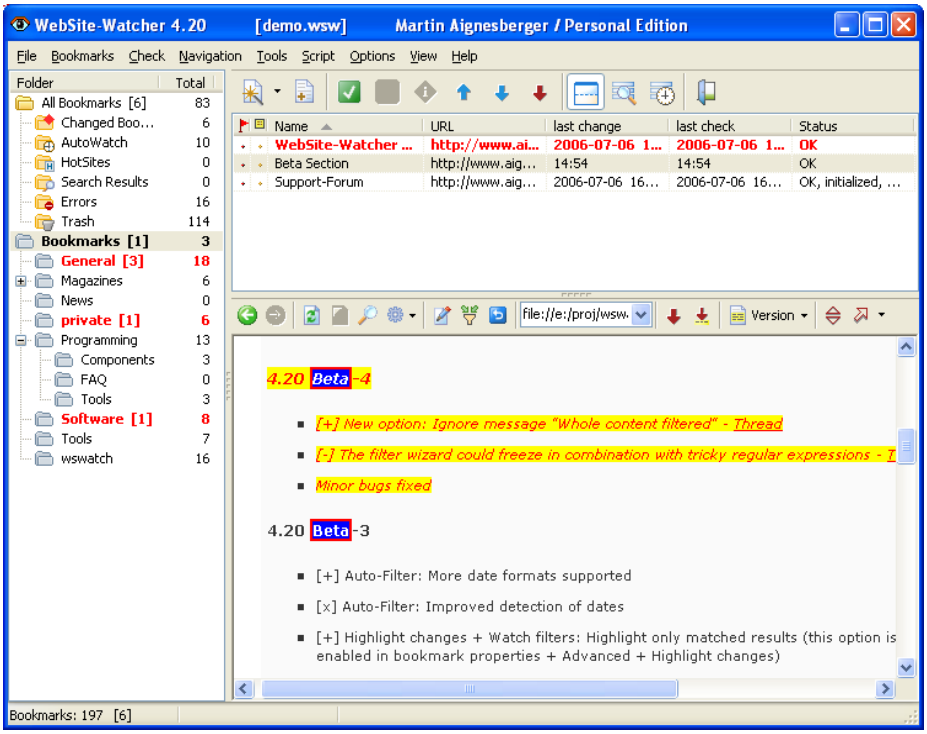
Free Malicious Code Databases:

1. Computer Associates Malcode Database
2. Symantec Malcode Database
3. Trend Micro Malcode Database
4. McAfee Malcode Database
5. F-Secure Malcode Database

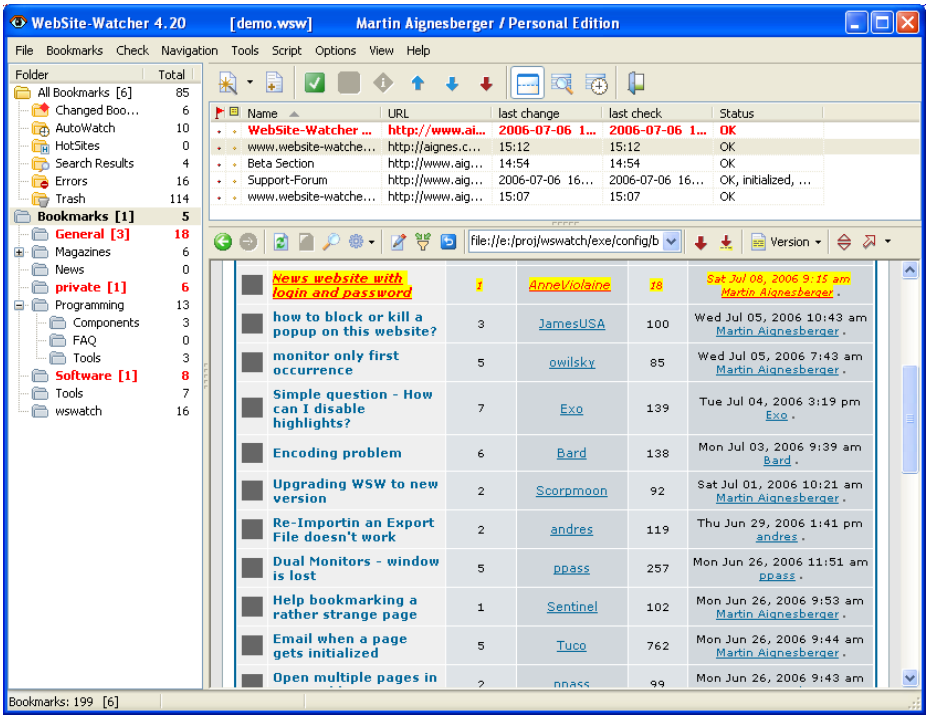
It is a very tedious task to individually monitor all the above security Web sites to gather the latest threat intelligence. To make this process much easier for the malware analysts, a monitoring software tool called WebSite-Watcher allows the malware analysts to monitor websites for updates and changes. By using this software tool, researchers will be able to use a single web site that provides alerts when content is updated.

WebSite-Watcher highlights all changes found in pages being monitored by malware analysts.

Below are snapshots that show how WebSite Watcher highlights changes.



Web page with highlighted changes, specified words (eg. "beta") are highlighted in blue.



Forum with new postings and replies.

Malware Collection:
Collecting malware is very crucial for identifying new and unknown internet security threats. There is one main problem facing security researchers which is obtaining and collecting malware quickly. In order to make the process of collecting malware quicker, we need to automate the malware Collection Process. This process starts with finding reliable malware collector tools and malware feeder source or malware contributors.

After intense research, we found various different open source software tools which could help malware analysts automate the complete process of capturing a malware binary, analyzing it, and finally tracking the associated botnet.

The First Strategy for automating malware collection is using the Nepenthes tool. Nepenthes is an easy solution for collecting worms and other autonomous spreading malware in a non-native environment like FreeBSD or Linux.

Nepenthes was developed by the German Honeynet Project. The main idea behind nepenthes is the emulation of vulnerable services. Instead of deploying a high-interaction honeypot with vulnerable services that can be exploited by autonomous spreading malware, this tool emulates the services. This reduces the risk of running a honeynet since nepenthes does not run a vulnerable service, an attacker can not fully compromise the honeypot. The attacking process will interact with emulation and thus we mitigate the risk involved.

Once malware analysts have downloaded a piece of malware, it is stored on the hard disk and never executed. So the honeypot is never infected with malware – something impossible with a high-interaction honeypot. This approach also leads to better scalability. The German Honeynet Project was able to run several thousand honeypots on just one physical machine.

According to the German Honeynet Project more than 15,000 malware binaries have been collected with the help of nepenthes. The tool proves to be very useful and several other nepenthes sensors have been deployed by members of the malware collection alliance. Also, many non-honeynet related organizations are using nepenthes on a daily basis. The collected malware consists of mostly bots, but there are also diverse other types of malware.

The second strategy is to encourage submissions of malware samples by other non- honeynet researchers or individuals such as the following dedicated malware collectors:

1. http://www.offensivecomputing.net
2. http://www.frame4.net/mdpro
3. http://vx.netlux.org
4. http://shadowserver.org/wiki

The following are free online malware scanners:

1. http://www.virustotal.com
2. http://virusscan.jotti.org
3. http://scanner.virus.org

The above scanners have proved to be very effective. They allow for submission to many AV vendors at once. Individuals need to be encouraged to scan suspicious malware files found in their effected systems using the above mentioned scanners to find out if the malware has been detected by other AV vendors. This strategy can be accomplished by dedicating a page for external users to submit unique malware for the USMA malware analysts to analyze.

What is expected from the USMA?

USMA needs to join the malware collection alliance by deploying nepenthes sensors and share malware samples. This will get USMA into the malware sharing business.

The above objective could be accomplished by simply deploying nepenthes sensors by the USMA to capture malware**.** By combining nepenthes with other open source tools such as automated Norman Sandbox, CWSandbox, Truman, etc USMA malware analysts will be equipped with a unique set of tools which will allow them to be on top of malware activity. So, they will not need to deploy full high interaction honeypots.

Next, we will discuss in more detail some of the open source tools mentioned above:

1. Nepenthes
2. Norman Sandbox and CWSandbox
3. Truman

Nepenthes:

According to the Nepenthes website, Nepenthes is a versatile malware collection sensor that emulates known and widespread Windows

vulnerabilities. It downloads malware trying to exploit these vulnerabilities. It then saves shellcode and binaries but never gets infected with them.

Nepenthes allows attackers, worms, bots, and other kinds of malware to attack it. This will lead to downloading the associate binary files and other payloads. The collected information by nepenthes sensor helps malware analysts identify possible attacks.

Nepenthes 2.0 is the newest version of nepenthes. The following changes to this latest version are as follow:

- ✓ ***module-honeytrap*:** a new module that ports the idea of honeytrap to nepenthes.
- ✓ ***submit-postgres*:** a module to collect binaries in a central database.
- ✓ Integration of CWSandbox in nepenthes via Norman submission.
- ✓ Some bug fixes and more enhancements to the existing modules.

Norman Sandbox and CWSandbox:
As discussed above, integrating Norman Sandbox and CWSandbox with Nepenthes will help to automate the complete process of capturing a malware binary, analyzing it, and finally tracking the associated botnet.

According to the Norman website, Norman Sandbox live automatically runs submitted malware files in a test environment. This test environment watches what the file does. Norman Sandbox live allows users to upload malware for analysis. Norman Sandbox reports provide useful information such as files and registry keys changed by malware, and other actions taken by the malware.

According to the Norman website, when a file is passed to the Norman Sandbox for scanning it is first tested using the malware signature files to see if it is known malware. If this malware turns out to be known by other AV vendors, then Norman labels it as known malware. If it is not found in Norman signature files, it is passed on to the Norman Sandbox where the file is executed in a secure and closed environment and then monitored for suspicious activities. If it is harmful, it is reported as an unknown malware.

According to the Norman Sandbox and CWSandbox websites, both sandboxes are not a pattern matching only virus scanners, the scanners can run files in their own internal sandbox, profile them, and guess

whether or not the behavior is malicious. So, security analysts don't need to reverse engineer the file to gain any further information; each sandbox will generate a clear and coherent report which can be emailed to security analysts.

The latest SVN version of nepenthes is able to submit binaries to both Norman Sandbox and CWSandbox. These tools could help USMA automate botnet detection and mitigation.

Truman:
According to Truman website, Truman is the "reusable unknown malware analysis net". It is an open-source behavioral malware analysis sandnet.

Joe Stewart, the person who came up with Truman, is overwhelmed with the amount of malware that needed analysis. He set out to build a system to automate many of the tasks done during malware analysis. Joe setup a windows box to infect, a linux box for analysis and a low interaction honeypot to emulate the vulnerable network; he finally came up with a tool called Truman.

Truman consists of a Linux boot image and a collection of scripts. Also provided is pmodump which is a Perl-based tool to reconstruct the virtual memory space of a process from a PhysicalMemory dump. With this tool, it is possible to avoid most packers to perform strings analysis on the dumped malware.

Truman can be used to build a "sandnet" for the USMA. This tool will help security analysts analyze malware in an isolated environment. Yet, it provides a virtual Internet for the malware to interact with. The biggest problem with not using VMs is the difficulty involved with constantly re-imaging machines so it can be used again.

Deploying Truman at the USMA will be the last step after successfully deploying the first tools (HoneyTIC, Nepenthes, Norman Sandbox, CWSandbox) and making sure that these tools are up and running.

Truman will be deployed as the last tool to automate the malware analysis process, leaving the malware analysts with only a little work to do in order to get an initial analysis of a piece of malware.

Summarizing the USMA malware collection process:
Malware capturing for analysis can be a little tricky. Fortunately, there are both open source and commercial products that can help. One of the

downfalls is getting a central system together to manage all of this. The following shows how USMA will tackle this process:

Identifying: HoneyTIC

Capture: Honeywall and Nepenthes sensors.

Analyze: Norman Sandbox, CWSandbox and Truman.

Alert: Notify the appropriate public and private organizations about newly discovered malware. Also, notify these organizations which AV vendors found the malware to be malicious.

2nd Phase (Investigating):

As mentioned above, malware analysts need to be focusing on the most current and up to date information and updates about any outbreak for any kind of malware. The plan here is to identify known and unknown malware in real-time.

After receiving a threat by the HoneyTIC, an in-depth investigation starts right away to find out if this threat is associated with any malware causing the threat being investigated.

The investigation phase could be accomplished by establishing a forum in the HoneyTIC website where security analysts' discussions and sample sharing takes place. Automated and non-automated daily, weekly and monthly collection sharing reports will be published. These reports are very critical when it comes to identifying new malware.

To make the investigation phase much easier, it is divided into three critical processes:

1. Accurate Prioritization
2. Early Warning to Emerging Threats
3. Code Analysis and Response

Accurate Prioritization:

- ✓ Studying and analyzing all intelligence gathered threats and newly discovered vulnerabilities and associated malware are very critical to be able to make the right decision and accurately prioritize the scale of the threat, vulnerabilities and malware discovered.

- ✓ Whenever a very distinctive malware has been found, it needs to be checked to see if it has been detected by other members of the

malware collection alliance or other common AV vendors such as VirusTotal.

- ✓ It is very critical when investigating a threat that malware analysts discuss and share private information in order to let analysts understand the presented threats. Understanding threats will enable them to detect a variety of known attack patterns, such as worms, denial of service attacks and other malicious activities that require instant responses.

Early Warning to Emerging Threats:

- ✓ Time is very sensitive when it comes to protecting national critical information assets against cyber-threats. Staying on top of the threat and vulnerability landscape requires the very best security intelligence. This threat security intelligence will help with the discovering new malware which could be the cause behind the threat being identified.

- ✓ The techniques and processes that will be implemented in the HoneyTIC website should deliver early warnings to emerging threats. Early warnings will provide organizations with more time to protect their critical information assets.

- ✓ The aim here is to identify and find known and unknown threats in real-time and make the security community aware of them. With collaboration with other security researchers, all unknown Internet activities will be analyzed by security experts to discover new threats, attack techniques and new and unique malware spreading the wild and exploiting new or unpatched vulnerabilities.

In this phase, in-depth investigation will be done if one of the following circumstances occurs:

- ✓ If new vulnerability has been released but not confirmed yet.
- ✓ If there is no sample captured by Malware collection tools, more investigation is needed to acquire a sample for analysis.
- ✓ Questionable scripts sent by private or individual contributors.
- ✓ …etc.

After the code or script has been analyzed, the USMA malware analysts have to decide if the threat level of the code is LOW, MEDIUM or HIGH. After identifying the threat level, this will lead the USMA malware analysts to perform any rapid response that might be needed.

Code Analysis and Response:
Collaboration among members of the malware collection alliance is very important when it comes to acquiring a malware sample for code analysis. Collaboration can be in the form of sharing and contributing malware collected with the help of nepenthes sensors or any other collection method by private sources and individuals.

After malware analysts acquire what's believed to be a unique malware sample by any source, this sample will be waiting for special process escalation to determine the uniqueness of it.

The following are the three types of escalations that may take place when analyzing any malicious code:

1. Vulnerability research
2. Behavioral Analysis
3. Reverse-engineering

Vulnerability research:
Zero-day attacks or critical vulnerability exploitation risks, such as a critical vulnerability exploit within a few days of the patch, qualify for this type of escalation. In this situation a lead researcher for the vulnerability will be assigned if available (preferably) and escalation occurs rapidly with minimal research prior to escalation by USMA Real-Time Response Team (RTRT). At a minimum the grounds for vulnerability escalation considerations must be established by RTRT prior to Labs escalation. E.g. how do you know it's likely related to this vulnerability and why is this a potential escalation situation.

Behavioral Analysis:
Whenever malicious code or any other suspicious binary has been captured by the USMA nepenthes sensors or any other contributing source, the malware analysts should start an immediate analysis of the malicious code. This could be done by validating the behavioral testing of that specific malicious code within a simulated operating system or networked environment.

When conducting behavioral analysis the following are the primary data that could be discovered:

- ✓ Copies of the script and/or code to be analyzed
- ✓ Binary strings dump of the primary code
- ✓ Files added, deleted, or modified by the primary code
- ✓ Annotated list of files, basic function and relationship to the incident

- ✓ Registry keys added, deleted, or modified by the primary code
- ✓ Ports opened by the primary code
- ✓ Packet captures of network traffic
- ✓ Win32 API calls
- ✓ Any other data pertinent to the analysis

Reverse-engineering:
if behavioral analysis of malware did not result in identifying what type of malware it is, malware analysts will then use all relevant raw behavioral data that has been identified as useful from the behavioral analysis when performing reverse engineering.

The following circumstances are when the USMA malware analysts should call or notify other members of the malware collection alliance or other point of contacts:

1. If after analyzing a malware sample, the malware analyst finds the malware to have interesting techniques that has never been seen in different malware.

2. If the nepenthes sensors watchdog(s) see activity in the sensors about a particular type of code, which has previously been reported on BAGLE, MYTOB, ETC.

3. If any of the above Security sites is reporting about a new method of malicious code exploitation being reported with insufficient data.

4. If high traffic affects the nepenthes sensors which may be associated with a particular reported malicious code variant.

If any of the above circumstances has been met, malware analysts can then start analyzing the code and after doing that they must document all the fingerprints found and write a clear and concise report. Next, they should send out an email to other researchers if the code is important enough to require the involvement of other security experts.

3rd Phase (Reporting):
When malware analysts have completed a rapid response they will package all related data and analysis files into a password protected zip file with the password "infected". The results will be disseminated as appropriate.

If the malware that has been captured is believed to exploit a very urgent un-patched vulnerability, a reporting process will take effect and the

appropriate people will be notified. For example, this malware will be reported to an anti-malware software vendor such as McAfee, Symantec, Panda, Sophos, Trend Micro to make signature for this malware so it can be detected.

It will also be a good idea to deliver a Daily, Weekly, and Monthly Honeynet Alliance Intelligence Gathering Report Summary describing the findings and any malware trends that are being seen.

To accomplish the reporting phase, the USMA needs to dedicate a special section in the HoneyTIC website for reporting. This reporting section will report on early warnings to emerging threats such as:

1. Report about a newly discovered vulnerability
2. Report an Incident found by the USMA malware analysts
3. Report about a new malware that exploits un-patched vulnerabilities
4. Report Phishing and spam
5. Report on whatever else is important

What will be deployed by the USMA?

- ✓ Building two identical Debian and Ubuntu Linux systems each running the default configuration of the latest version of Nepenthes.

- ✓ After deploying the above nepenthes sensors, USMA malware analysts will monitor both sensors for one month. The gathered malware of the two sensors will be compared and a short report will be created and submitted to Colonel Dodge for review.

- ✓ Deploy nepenthes sensors on a T1 line or other dedicated connections.

- ✓ Deploy a malware collection repository database to store captured malware by USMA nepenthes sensors. This database will serve as a central repository and will be used to store captured malware.

It is very important for the USMA malware analysts to create a master database for gathering all hacker activities. This database could be used as a service where USMA honeynet researchers and malware analysts can access and acquire all kinds of malware, network traffic or any suspicious scrip acquired from private sources. This will help malware analysts quickly analyze suspicious files and aid in the quick detection of

hacking malicious activities and all kinds of malware detected by antivirus companies.

Conclusion

Using security open source tools such as nepenthes sensors, Norman Sandbox, CWSandbox, and Truman will help security experts indicate the next malware global risk. It is very clear that deploying nepenthes sensors can be used to capture malware, alert security and malware analysts about a network compromise, and assist in containing and removing the infection.

By working very close to other malware collection alliance members all over the world, the USMA malware analysts could share information and valuable findings with other alliances. This will help the USMA nepenthes project and other alliances focus on network attacks and keep looking for the next possible global risk. By identifying the next possible global risk, this will aid in reporting the findings and notifying the appropriate government and security organizations.

References:

(n.d.). *Nepenthes - finest collection -*. Retrieved September 10, 2006 from Web site: http://nepenthes.mwcollect.org

(n.d.). *Nepenthes - Documentation*. Retrieved October 9, 2006 from Web site: http://nepenthes.mwcollect.org/documentation:readme

Werner, T. (n.d.). *HoneyTrap – trap attacks against tcp services*. Retrieved September 12, 2006 from, Web site: http://honeytrap.sourceforge.net/start.html

(n.d.). *Automated Malware Analysis with Instant Results*. Retrieved September 15, 2006 from, Web site: http://www.norman.com/Product/Sandbox-products/en

University of Mannheim (n.d.). *CWSandbox: Automatic Behaviour Analysis of Malware*. Retrieved September 15, 2006 from Laboratory for Dependable Distributed Systems Web site: http://www.cwsandbox.org

Stewart, J. (n.d.). *Truman - The Reusable Unknown Malware Analysis Net* . Retrieved October 20, 2006 from, Web site: http://www.lurhq.com/truman

How to Install Nepenthes 0.2.0 on Ubuntu 6.06 Server Edition

The following steps show how to install Nepenthes 0.2.0 on Ubuntu.

This document may also work well with Debian too.

I installed Ubuntu 6.06 Dapper Drake Server Edition.

I created a **user account** and set the **time zone**, but other than that, **all options** were set to their **defaults**. This distro finishes installation with **NO ports open** at all. Excellent!

Host name: `localhost`

User login: `user1`
Password: `user1`

For prerequisites install the following packages:

- `sudo apt-get install` **`libcurl3-dev`**
- `sudo apt-get install` **`libmagic-dev`**
- `sudo apt-get install` **`libpcre3-dev`**
- `sudo apt-get install` **`libadns1-dev`**
- `sudo apt-get install` **`libpcap0.8-dev`**
- `sudo apt-get install` **`iptables-dev`**
- `sudo apt-get install` **`autoconf`**
- `sudo apt-get install` **`automake1.9`**
- `sudo apt-get install` **`autotools-dev`**
- `sudo apt-get install` **`libtool build-essential patch`**

Go somewhere predictable:

```
sudo chmod o+w /usr/src
cd /usr/src
mkdir nepenthes
cd nepenthes
```

To get Nepenthes I did the following:

`wget` http://umn.dl.sourceforge.net/sourceforge/nepenthes/nepenthes-0.2.0.tar.bz2

```
tar xvjf nepenthes-0.2.0.tar.bz2
```

Check the current directory nepenthes got unzipped to. Type `ls`:
Results:

```
Nepenthes-0.2.0 nepenthes-0.2.0.tar.bz2
```

Nepenthes-0.2.0 is the directory we are looking for. Type `cd nepenthes-0.2.0`.

Then to configure and build it:

```
./configure --prefix=/opt/nepenthes
```

```
make
sudo make install
```

I liked how the **configs** were set up to begin with. **Simple logging**, **feigning vulnerability to everything it can**, and **saving the samples to disk**, but I also wanted reports emailed to me on the samples received, so I edited (with **nano**) **nepenthes.conf**:

```
nano /opt/nepenthes/etc/nepenthes/nepenthes.conf
```

Look for the line that contains:

```
// “submitnorman.so”, “submit-norman.conf”,  ””
```

And remove the “//” from the beginning. Save and exit.

Now let’s run it:

```
sudo bin/nepenthes
```

Malware Analysis

WHY?
To provide information regarding:

- ***Mitigation:*** Strategies that prevent getting infected.
- ***Detection:*** Strategies that detect already infected systems.
- ***Removal:*** Strategies that remove an identified infection.

HOW?

- Analysis steps that are easier and faster to perform typically bring less information.

- If neither threat assessment nor other inquiries dictate a certain action, start with the easy steps and walk your way up until you collect enough information.

- What does this mean: "enough information"? Always know what kind of information is needed and choose methods accordingly.

- Depending on the code, analysis of network behavior might be more complex and/or might more results compared to a strings analysis.

- None of the above analysis methods will be able to determine the initial attack vector for codes. E.g. if a Trojan was heavily spammed via email, if a website hosting the malicious code browser exploits, etc. Additional intelligence is necessary to analyze the outbreak situation.

0. AV Detection

AV scan is quick and the result might determine that the code is already known and reported on. Or, it might show that the code is nearly undetected and together with prevalence information this might result in updated risk rating.

The following multi AV scanner can be used:

- VirusTotal: http://www.virustotal.com
- CWSandbox: www.cwsandbox.org
- Jotti.org: http://virusscan.jotti.org
- Virus.org: http://scanner.virus.org
- Norman: http://sandbox.norman.no/live.html

1. Malcode's Behavior

This will determine how a malicious code is behaving once it has infected a system. It will determine how this code behaves in the given circumstances. This does not necessarily (and typically it is not) mean that the code always under all circumstances behaves this way. Furthermore, the code might have extra functionality that is not shown in the lab environment due to whatever reasons.

We can observe the actions of the running malware on the host itself (file system, memory, and registry) or on the network (network traffic).

1.1 Host Behavior

Classic behavioral analysis includes registry changes, file system changes, mutexes, and listening ports. This analysis can be done with a Host IDS (file integrity checker such as InCtrl or InstallWatch), Process explorer, and fport (or netstat).

All this behavior on the local host can be concealed using rootkit techniques. Always keep this in mind and keep it open as an option when dealing with results that don't seem to make sense.

1.2 Network Behavior

If the system is designed correctly, the code will show close to its full potential. Capturing the network traffic in such an environment will deliver excellent detection information. In case of network worms it will also deliver mitigation information.

The ITOC Malware Lab can be used to provide the appropriate environment and the network traffic can be captured with Snort and Ethereal.

Be aware that one aspect of network functionality will not be captured: listening ports on the infected hosts!

2. Malcode's Full Potential (Embedded Functionality)

2.1 Guessing the included Functionality (From embedded strings)

From looking at the strings embedded in an executable and the library functions that are being utilized one can draw some good conclusions about what the code is intended to do. IRC contact info as well as download URLs can oftentimes be found here.

Be aware that this is just a guess and could easily be exploited by malware authors. Use this wisely and with backup info from behavioral analysis.

2.2 Full Reverse Code Engineering (RCE)

Some malware require analysts to perform RCE and this may take hours and days to complete and requires deep knowledge of the underlying programming language used, operating system, assembly language, byte code, etc.

3. Malicious Code Analysis (Code Handling Procedures):

3.1 As described above, scan each Malicious Code through any of the following Multi Scanners:

- VirusTotal: http://www.virustotal.com
- CWSandbox: www.cwsandbox.org
- Jotti.org: http://virusscan.jotti.org
- Virus.org: http://scanner.virus.org
- Norman: http://sandbox.norman.no/live.html

3.2 Check to see if the malware you want to analyze has been detected by any AV vendor by using any of the above Web sites. If the malware is not detected by any AV vendor, go a head and start performing Code Behavior Analysis as mentioned below:

4. Installing and Running the VMWare:

4.1 Install VMware-workstation and get the latest VMWare image from Ron or create your own one.

4.2 Go to My Computer, open My Virtual Machines folder and place VMWare image there.

4.3 Double Click on the VMware Player icon on the Desktop.

4.4 Another Window will open. Double click on **ITOC Box** to fire up the Virtual Operating System.

5. Running Malicious Code and Other Code Analysis Tools:

5.1 It is always recommended to copy the code you want to analyze to the VMware Desktop SHARE directory and run it from there.

5.2 The purpose of InCtrl5 tool is to find the new added File Name(s), File Size(s) and Registry Key(s) after executing the Malicious Code we want to analyze:

- Run InCtrl5 to record the system pre-installation Snapshot.
- Execute the Malicious Code.
- Run InCtrl5 to receive a report.

5.3 Create a folder in the SHARE directory and Name it as:
- "Results-the name of the code.exe"

5.4 After running InCtrl5, save the following 2 files to the results folder in the SHARE directory:

- .TXT
- .CSV

5.5 .CSV file contains all the dropped files. After identifying the dropped files, save resulted files to the results folder in the SHARE directory.

5.6 Run RootkitRevealer to find if there are any hidden files using RootKits.

5.7 Run Process Explorer to find running processes and any Mutex(s).

5.8 After running Process Explorer, click on the file(s) being dropped by the code and look at Mutenet to find if there is any Mutex.

5.9 Run Fport and NetStat to find any Open Ports.

5.10 After running Fport.exe and NetStatexe, compare the ports opened to the Fport.bat and NetStat.bat. By doing this you will know what port(s) opened by the code you are running.

6. How to acquire Malicious Code?

The best way to acquire Malicious Code is by using the **"**wget**"** command in Ubuntu.

Notes:

- Sometimes the added or dropped file has a folder icon. This is just to trick the victim to click on it.
- C:\\Windows\prefetch\ **.pf** → these are normal files created by the OS
- C:\\Windows\preflib\ **.dat** → these are sometimes normal files and sometimes not.
- The Taskmanager has been deactivated because of the code.
- The [path name]**.tmp** file is being used and we found that when we used the InCtrl Helper.

Malware Types

The following defines the major types of known malware:

- Trojan Horse
- Worm
- Virus
- Adware
- Spyware

Trojan (n.) A computer program that appears to be useful but that actually does damage.

Trojans spread when people are lured into opening a program because they think it comes from a legitimate source. Trojans can also be included in software that you download for free. Never download software from a source that you don't trust. A Trojan horse can technically be defined as malicious software that does not replicate itself.

Different types of Trojans:

- **Backdoors**- allow backdoor access to the compromised computer
- **Keyloggers**- log keystrokes in an attempt to obtain passwords and other information
- **Downloaders**- download additional components for added functionality or updates
- **Proxy**- turn the computer into a host for relaying potentially illegal activities
- **Bot**- turn the computer into a zombie for DDOS attacks or other purposes

---

Worm (n.) Self propagating malicious code. A worm generally spreads without user action and distributes complete copies (possibly modified) of itself, some of which only exist in memory. A worm can consume memory or network bandwidth, thus causing a computer to stop responding. Because worms don't need to travel via a "host" program or file, they can also tunnel into your system and allow somebody else to

take control of your computer remotely. Recent examples of worms included the Sasser worm and the Blaster worm. A worm is malicious software that replicates itself without infecting a host.

Different types of Worms:

- **E-mail** - spreads via e-mail applications and over e-mail.
- **Network** - uses the network, scanning, and exploitation of vulnerabilities for propagation.
- **P2P** - uses popular peer-2-peer networks for propagation, often masking itself as non malicious files.
- **IRC** - spreads over IRC, using DCC sends and or taking advantage in protocols/vulnerabilities.

Virus (n.) Code written with the express intention of replicating itself.
A virus attempts to spread from computer to computer by attaching itself to a host program. It may damage hardware, software, or information. A true virus does not spread without human action to move it along, such as sharing a file or sending an e-mail. A virus can technically be defined as malicious software that replicates itself, infecting a host.

Adware (n.) Legitimate and non legitimate marketing programs, capable of delivering advertisements to a user. Adware is functionality added to applications to display advertisements to the end user. Adware is considered a legitimate alternative offered to consumers who do not wish to pay for software. The advertisements usually run in a small section of the software interface or as a pop-up ad box on your desktop. In many cases, adware is a legitimate revenue source for companies who offer their software free to users. In other cases, the adware can be an advertising Trojan of sorts, or install itself without the users permission. This type of attack can be a sort of DOS, by constantly bombarding the user with advertisements. This type of bombardment can often interfere with a users work performance and be a serious annoyance.

Spyware (n.) Applications that monitor or track a computer users actions for potentially malicious or marketing purposes. Some freeware applications which contain adware do track surfing habits in order to serve ads. When the adware becomes intrusive like this, then we move it in the spyware category. Spyware is considered a malicious program and is similar to a Trojan horse in that users unwittingly install the product when they install something else.

Spyware works like adware but is usually a separate program that is installed unknowingly when you install another freeware type program or application. Once installed, the spyware monitors user activity on the Internet and transmits that information in the background to someone else. Spyware can also gather information about e-mail addresses and even passwords and credit card numbers.

Because spyware exists as independent executable programs, they have the capability to monitor your keystrokes, scan files on the hard drive, snoop other applications, such as chat programs or word processors, install other spyware programs, read cookies, change the default home page on the Web browser, while consistently relaying this information back to the spyware author who will either use it for advertising and marketing purposes or sell the information to another party.

Malware Analysis Guidelines

Distribution:

- Self-Replication
  - Virus
    - File-Infecting Virus
      - Types of files targeted
    - Macro Virus
  - Worm
    - Mass-Mailer (Own engine/MUA)
    - P2P
      - Open Shares
      - Password protected shares
      - Specific folder names/types
    - Network
      - Network scanning characteristics
        - Linear
        - Random
        - Specific IP ranges
    - IM
      - IM application used
      - Method of delivery
        - URL
        - Auto-propagating (exploit or otherwise)
- Non-SR Initial Infection Vector
  - seeded via e-mails or IM
  - posted on websites
  - immediate file-exchange
- Polymorphism (changes in size, packing, padding)

Target:

- Platform
  - OS (e.g. Win32, Linux, Java, Symbios, etc.)
  - Service (e.g. phpBB, IIS, etc.)
- Audience
  - Is attack/payload targeted at any of the following:
    - IP range
    - Domains

Exploitation:

- Type of Exploit
  - OS (IR#)
  - Application (IR#)

    - Web Browser (IR#)
- Public
- Patched (since when?)

Payload:

C-IA Triad Attack

- Information Stealer
    - What types of data or files are compromised?
    - Does the code hook into browser for data theft?
    - If so which browser, FF, MSIE, or both?
    - Is the code proxy aware?
    - Does it steal specific file types?
    - Does it use the key log buffer or capture screenshots?
    - How is the information sent out?
    - What institutions are targeted?
    - Where is the info sent?
- Information Destructor
    - What data is affected, file types, data type?
    - How is the data being affected, deleted overwritten, altered?
    - Is the data encrypted?

Control

- Backdoor
    - Ports
    - Protocol
    - Functionality
    - notify
- DoS
    - Port
    - protocol
    - type of DoS
    - target URL/IP address
    - schedule
- Spammer
    - msg-characteristics
    - own-engine/MUA
    - uses local SMTP or remote hardcoded list
    - does it inject data into SMTP steam
- Proxy
    - Port
    - protocol

Staged Attack

- Downloader
    - URL

        - port
        - protocol
        - file attributes
    - Dropper
        - attributes of all files dropped
    - Security Software Retaliation
        - What products are affected?
        - Files deleted?
        - Processes terminated?
    - Lower Security Settings
        - What's being changed?
        - What's the impact?
    - Rootkit
        - what processes, files, directories or ports are being concealed
    - Hacktool
        - describe functionality
- Combo Malware
    - IRC Bot
        - Server
        - Protocol
        - Port
        - Channel
        - User
        - Password
- Questionable/Unwanted
    - Clicker /Redirector
        - Target
        - conditions
    - Adware
        - Technique used [popup, etc.]
        - product/vendor advertised
    - Joke Program
        - Explain what the program does, functionality, display/sound changes

Conditional & Timing Components:

- Propagation
    - Time
    - Date
    - Event driven, if so describe event
- Payload execution
    - Time
    - Date
    - Event driven, if so describe event

Threat Assessment:

- Initial Infection Vector
  - Is exploitation involved?
  - What environment supports the infection vector?
- Environmental
  - What environments support the initial delivery method?
  - Does it run under restricted privileges?
- Propagation
  - Is the code self-replicating? If yes, what medium is used?
  - Is exploitation involved?
- Payload

Detection:

- Snort alerts triggered
- Event viewer changes
- Network traffic generated, type, protocol and port used

Malware Naming Convention Standard

The following shows the best way to name malware and keep it organized:

.X

.A	.B	.C	.D	.E	.F	.G	.H	.I	.J	.K	.L	.M	.N	.O	.P	.Q	.R	.S	.T	.U	.V	.W	.X	.Y	.Z

.XX

.AA	.AB	.AC	→	.AZ	→	**.BA**	.BB	.BC	→	.BZ	→	**.CA**	.CB	.CC	→	.CZ	→	**.ZZ**

.XXX

.AAA	**.ABA**	**.ACA**	→	**.AZA**	→	**.BAA**	**.CAA**	**.DAA**	→	**.ZAA**	→	**.ZZA**
.AAB	.ABB	.ACB		.AZB		.BAB	.CAB	.DAB		.ZAB		.ZZB
.AAC	.ABC	.ACC		.AZC		.BAC	.CAC	.DAC		.ZAC		.ZZC
.AAD	.ABD	.ACD		.AZD		.BAD	.CAD	.DAD		.ZAD		.ZZD
.AAE	.ABE	.ACE		.AZE		.BAE	.CAE	.DAE		.ZAE		.ZZE
.AAF	.ABF	.ACF		.AZF		.BAF	.CAF	.DAF		.ZAF		.ZZF
.AAG	.ABG	.ACG		.AZG		.BAG	.CAG	.DAG		.ZAG		.ZZG
.AAH	.ABH	.ACH		.AZH		.BAH	.CAH	.DAH		.ZAH		.ZZH
.AAI	.ABI	.ACI		.AZI		.BAI	.CAI	.DAI		.ZAI		.ZZI
.AAJ	.ABJ	.ACJ		.AZJ		.BAJ	.CAJ	.DAJ		.ZAJ		.ZZJ
.AAK	.ABK	.ACK		.AZK		.BAK	.CAK	.DAK		.ZAK		.ZZK
.AAL	.ABL	.ACL		.AZL		.BAL	.CAL	.DAL		.ZAL		.ZZL
.AAM	.ABM	.ACM		.AZM		.BAM	.CAM	.DAM		.ZAM		.ZZM
.AAN	.ABN	.ACN		.AZN		.BAN	.CAN	.DAN		.ZAN		.ZZN
.AAO	.ABO	.ACO		.AZO		.BAO	.CAO	.DAO		.ZAO		.ZZO
.AAP	.ABP	.ACP		.AZP		.BAP	.CAP	.DAP		.ZAP		.ZZP
.AAQ	.ABQ	.ACQ		.AZQ		.BAQ	.CAQ	.DAQ		.ZAQ		.ZZQ
.AAR	.ABR	.ACR		.AZR		.BAR	.CAR	.DAR		.ZAR		.ZZR
.AAS	.ABS	.ACS		.AZS		.BAS	.CAS	.DAS		.ZAS		.ZZS
.AAT	.ABT	.ACT		.AZT		.BAT	.CAT	.DAT		.ZAT		.ZZT
.AAU	.ABU	.ACU		.AZU		.BAU	.CAU	.DAU		.ZAU		.ZZU
.AAV	.ABV	.ACV		.AZV		.BAV	.CAV	.DAV		.ZAV		.ZZV
.AAW	.ABW	.ACW		.AZW		.BAW	.CAW	.DAW		.ZAW		.ZZW
.AAX	.ABX	.ACX		.AZX		.BAX	.CAX	.DAX		.ZAX		.ZZX
.AAY	.ABY	.ACY		.AZY		.BAY	.CAY	.DAY		.ZAY		.ZZY
.AAZ	.ABZ	.ACZ		.AZZ		.BAZ	.CAZ	.DAZ		.ZAZ		.ZZZ

.XXXX

.AAAA	**.AABZ**	**.AACZ**	**.AADZ**	**.AAEZ**	**.AAFZ**	**.AAGZ**	**.AAHZ**	**.AAIZ**	**.AAJZ**	**.AAKZ**	**.AALZ**	**.AAMZ**
.AAZZ	.ABZZ	.ACZZ	.ADZZ	.AEZZ	.AFZZ	.AGZZ	.AHZZ	.AIZZ	.AJZZ	.AKZZ	.ALZZ	.AMZZ
.AZZZ	.BZZZ	.CZZZ	.DZZZ	.EZZZ	.FZZZ	.GZZZ	.HZZZ	.IZZZ	.JZZZ	.KZZZ	.LZZZ	.MZZZ