zanshinsecurity.com

Defending against Zero-Day attacks

Bob Mahoney

Zanshin Security, LLC

MIT Security Camp August 15-16, 2006

Presentation focuses on the Incident Response relevance of a research project Zanshin did for Verdasys, Inc. (http://verdasys.com) earlier this year.

Project considered the role their “Digital Guardian” product might play in a Defense-in-Depth strategy, and gauging the effectiveness against Zero-Day attacks.

I lifted this image from http://web.bvu.edu/students/pickbry/others.htm I sent mail asking for permission, or a pointer to someone who could give permission. no answer yet. And you can see, I’m using it anyway.

Somewhere, an intellectual property lawyer just got his wings.

Defending against Zero-Day attacks

Digital Guardian and Defense-in-Depth

Bob Mahoney

Zanshin Security, LLC

MIT Security Camp August 15-16, 2006

Presentation focuses on the Incident Response relevance of a research project Zanshin did for Verdasys, Inc. (http://verdasys.com) earlier this year.

Project considered the role their “Digital Guardian” product might play in a Defense-in-Depth strategy, and gauging the effectiveness against Zero-Day attacks.

I lifted this image from http://web.bvu.edu/students/pickbry/others.htm I sent mail asking for permission, or a pointer to someone who could give permission. no answer yet. And you can see, I’m using it anyway.

Somewhere, an intellectual property lawyer just got his wings.

The following presentation contains graphic references to commercial software, and may be inappropriate for our younger viewers.

Yes, this is a commercial product.

But like the other tips and techniques we share at Camp, the content or ideas might be useful to some people.

Most of us have some commercial software in play already anyway, but I want to make clear where I'm coming from:

Not here to sell you a product. Don't know what it costs. I have no personal interest in the matter should you buy it.

The following presentation contains graphic references to commercial software, and may be inappropriate for our younger viewers.

[adults win]

Yes, this is a commercial product.

But like the other tips and techniques we share at Camp, the content or ideas might be useful to some people.

Most of us have some commercial software in play already anyway, but I want to make clear where I'm coming from:

Not here to sell you a product. Don't know what it costs. I have no personal interest in the matter should you buy it.

Defense-in-Depth is a strategy of implementing multiple layers of defensive capabilities around a protected item or system. This reduces the reliance on any one mechanism, and adds opportunities to block or reduce the damage from attacks against previously unknown vulnerabilities.

Malware is a general term for “malicious software”. Everyone agrees a virus is malware, but near the edges live things like “beneficial” worms and DRM mechanisms such as the now famous “Sony rootkit”.

Zero-day exploits are released on the same day the vulnerability becomes known to the public. The term derives from the number of days between a public advisory and the release of an exploit. The term 'zero-day exploits' is sometimes used to indicate publicly known exploits for which patches have not yet been made available.

Defense-in-Depth

Defense-in-Depth is a strategy of implementing multiple layers of defensive capabilities around a protected item or system. This reduces the reliance on any one mechanism, and adds opportunities to block or reduce the damage from attacks against previously unknown vulnerabilities.

Malware is a general term for “malicious software”. Everyone agrees a virus is malware, but near the edges live things like “beneficial” worms and DRM mechanisms such as the now famous “Sony rootkit”.

Zero-day exploits are released on the same day the vulnerability becomes known to the public. The term derives from the number of days between a public advisory and the release of an exploit. The term 'zero-day exploits' is sometimes used to indicate publicly known exploits for which patches have not yet been made available.

Defense-in-Depth

Malware

Defense-in-Depth is a strategy of implementing multiple layers of defensive capabilities around a protected item or system. This reduces the reliance on any one mechanism, and adds opportunities to block or reduce the damage from attacks against previously unknown vulnerabilities.

Malware is a general term for “malicious software”. Everyone agrees a virus is malware, but near the edges live things like “beneficial” worms and DRM mechanisms such as the now famous “Sony rootkit”.

Zero-day exploits are released on the same day the vulnerability becomes known to the public. The term derives from the number of days between a public advisory and the release of an exploit. The term 'zero-day exploits' is sometimes used to indicate publicly known exploits for which patches have not yet been made available.

Defense-in-Depth

Malware

Zero-Day

Defense-in-Depth is a strategy of implementing multiple layers of defensive capabilities around a protected item or system. This reduces the reliance on any one mechanism, and adds opportunities to block or reduce the damage from attacks against previously unknown vulnerabilities.

Malware is a general term for “malicious software”. Everyone agrees a virus is malware, but near the edges live things like “beneficial” worms and DRM mechanisms such as the now famous “Sony rootkit”.

Zero-day exploits are released on the same day the vulnerability becomes known to the public. The term derives from the number of days between a public advisory and the release of an exploit. The term 'zero-day exploits' is sometimes used to indicate publicly known exploits for which patches have not yet been made available.

Digital Guardian

Digital Guardian

Central server manages intelligent agents

Digital Guardian

Central server manages intelligent agents

Agents enforce compliance with policy

Digital Guardian

Central server manages intelligent agents

Agents enforce compliance with policy

Server provides alert and audit functions

Digital Guardian

Central server manages intelligent agents

Agents enforce compliance with policy

Server provides alert and audit functions

System functions as a “Reference Monitor”

The Reference Monitor model:

Provides mandatory enforcement of security policies regarding all *user*, *program*, or *data* transactions.

"The Reference Monitor watches what other processes do and, where necessary, intervenes; otherwise it is, like the very best security products, entirely invisible and entirely inescapable. A good conscience is like that, too; just as no one wants to live with people who do not have a conscience it is now time to say that no computer that has its hands on valuable bits should not have a reference monitor."

From a Verdasys Whitepaper, "Defending in Depth" http://verdasys.com/pdf/did.pdf

http://www.craigchamberlain.com/ has a link to a paper by Craig Chamberlain, Donato Bucella, Daniel Geer, Sc.D. Presented at DHS Science & Technology 2005:

"A Host Reference Monitor Approach to the Problem of Human and Programmatic Insider Threat to Computer Information Systems"

The Reference Monitor model:

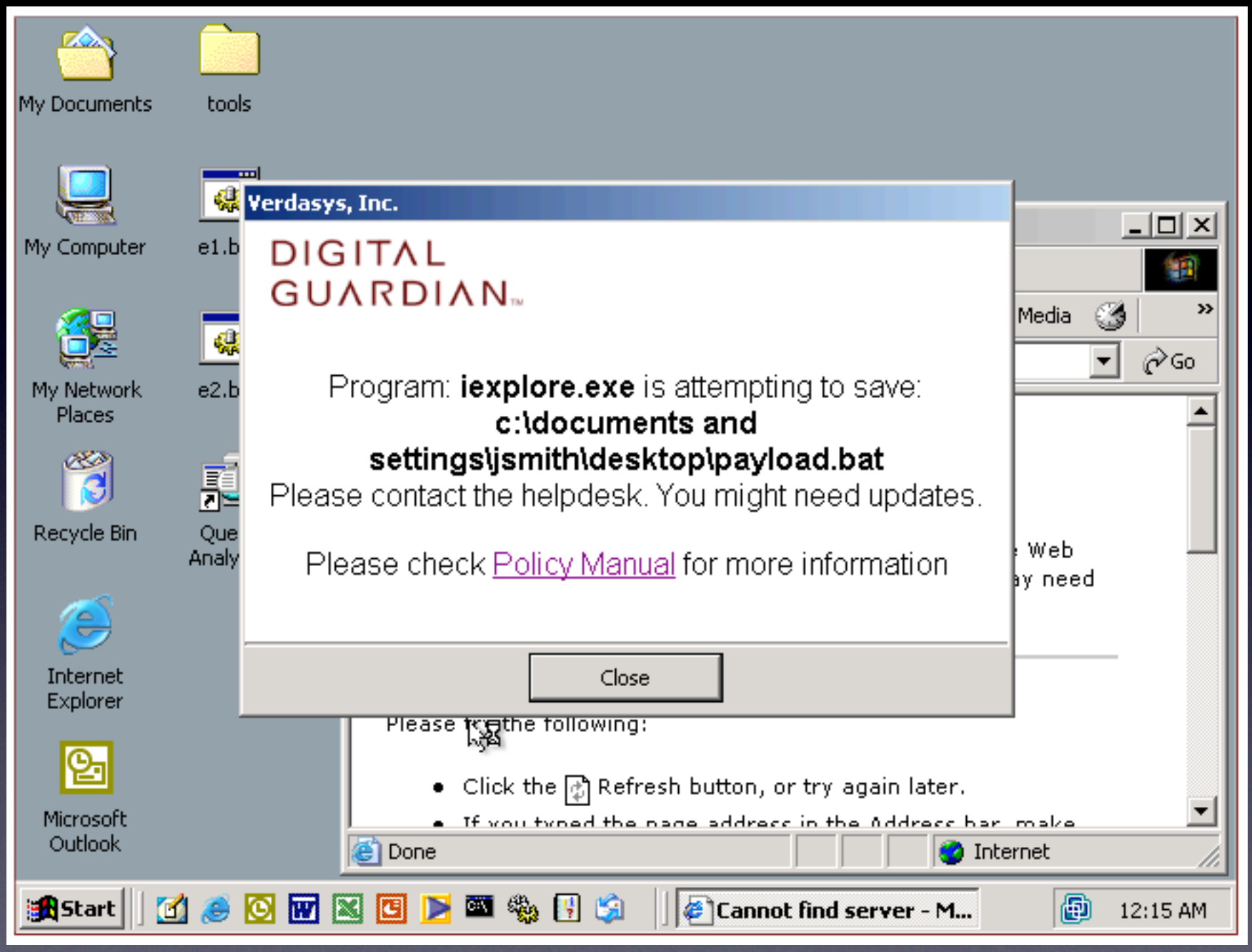
Put another way, the system functions much as a conscience would in an honest person.

“The Reference Monitor watches what other processes do and, where necessary, intervenes; otherwise it is, like the very best security products, entirely invisible and entirely inescapable. A good conscience is like that, too; just as no one wants to live with people who do not have a conscience it is now time to say that no computer that has its hands on valuable bits should not have a reference monitor.”

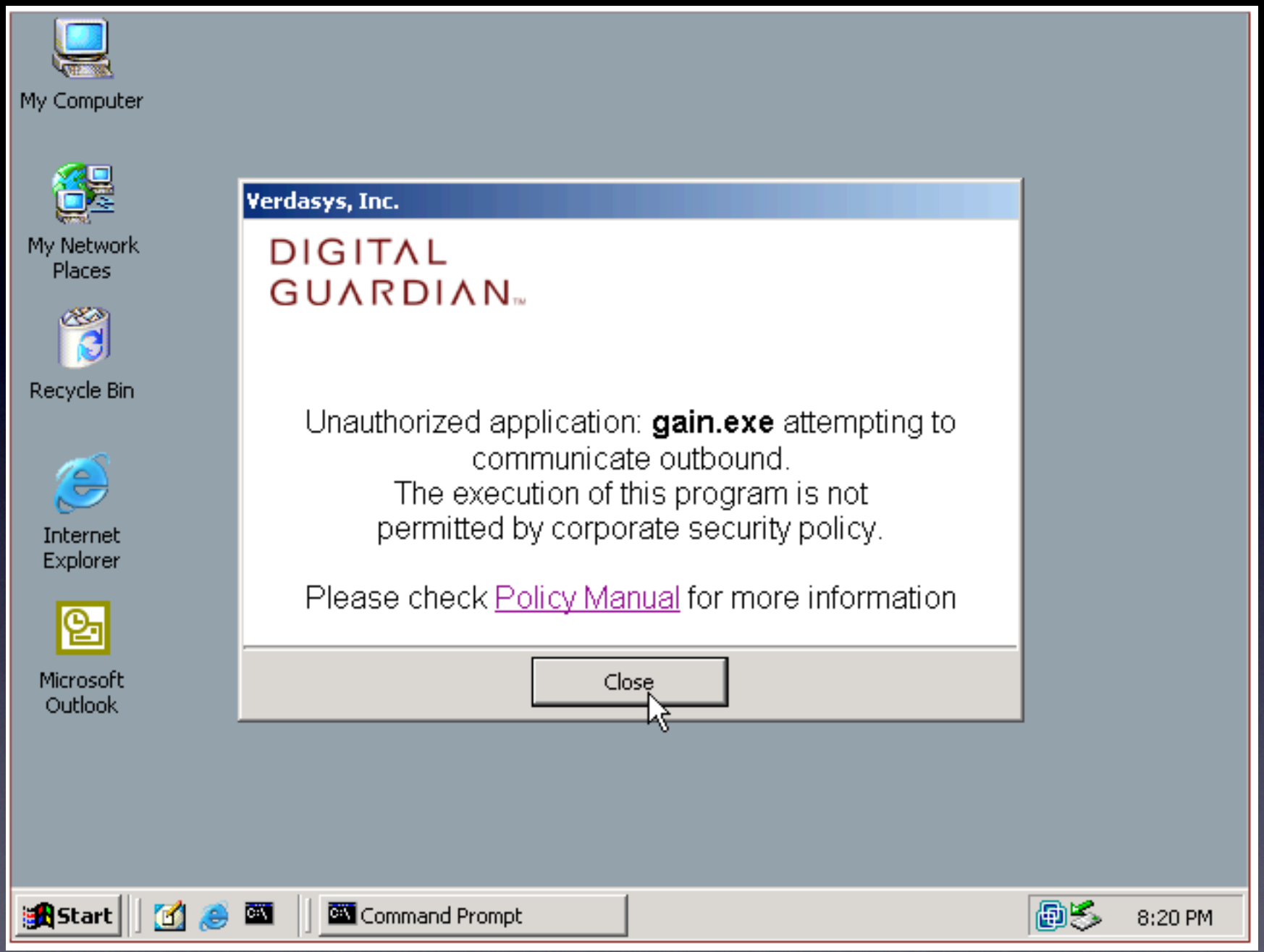
From a Verdasys Whitepaper, “Defending in Depth” http://verdasys.com/pdf/did.pdf

http://www.craigchamberlain.com/ has a link to a paper by Craig Chamberlain, Donato Bucella, Daniel Geer, Sc.D. Presented at DHS Science & Technology 2005:

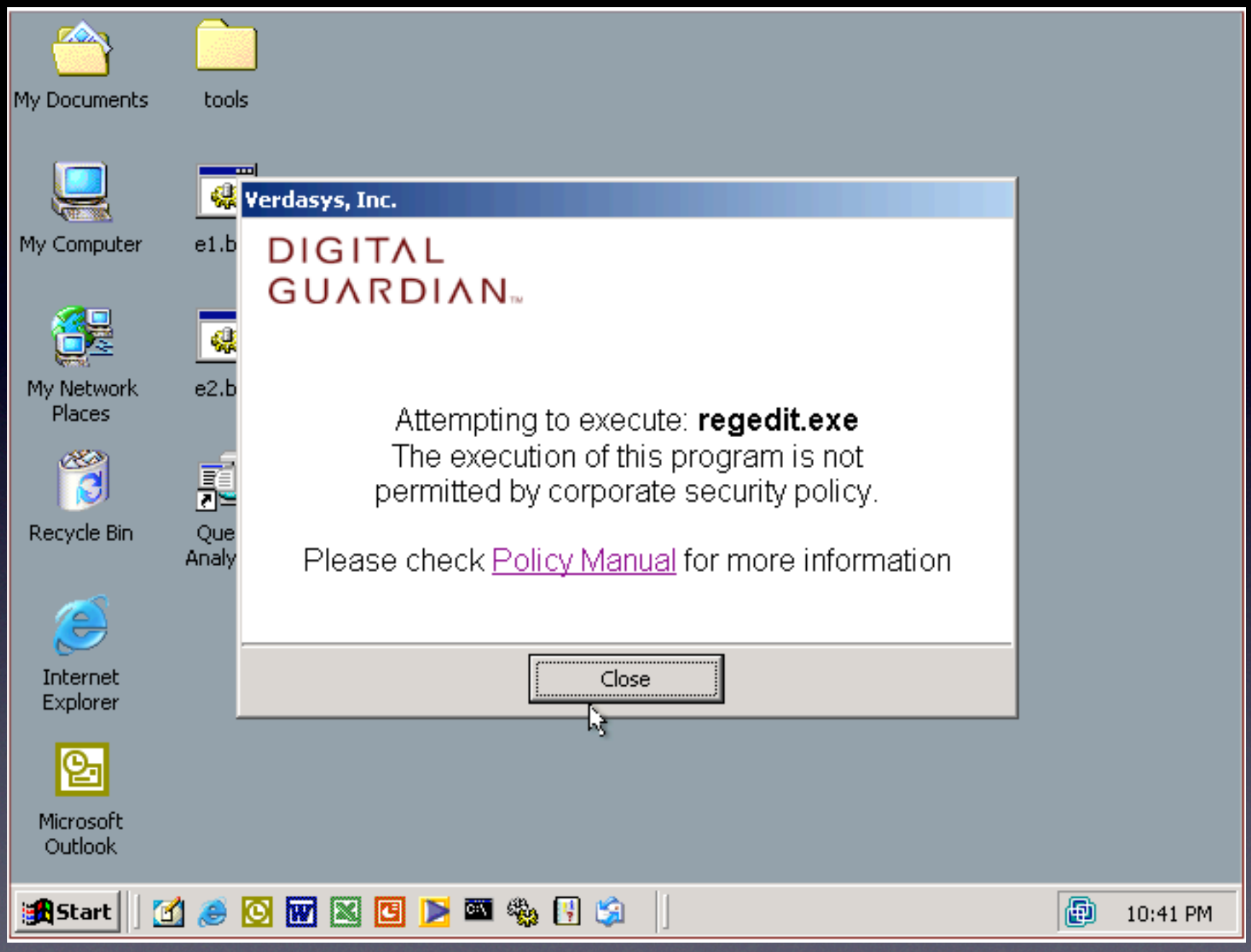
“A Host Reference Monitor Approach to the Problem of Human and Programmatic Insider Threat to Computer Information Systems”



Screen shots grabbed from this Flash demo: http://verdasys.com/demos/def_in_depth



Screen shots grabbed from this Flash demo: http://verdasys.com/demos/def_in_depth



Screen shots grabbed from this Flash demo: http://verdasys.com/demos/def_in_depth

Our Mission:

A primary goal of an organization's IT security function is to make sure that the IT assets are available for their intended use, and can be relied upon. Other goals such as preventing fraud, intrusion, or other misuse are critical, of course, but IT assets are a tool, and the tool has to function as intended for their to be any basic value.

We need to protect corporate financial records, but the primary role of these systems is to manage and support the financial operations of the organization.

When security appears to trump functionality, it's likely that the real function of the asset is security-related. (military, life-safety, etc)

Our Mission:

Maintain the integrity, availability, and security of organizational IT assets.

A primary goal of an organization's IT security function is to make sure that the IT assets are available for their intended use, and can be relied upon. Other goals such as preventing fraud, intrusion, or other misuse are critical, of course, but IT assets are a tool, and the tool has to function as intended for their to be any basic value.

We need to protect corporate financial records, but the primary role of these systems is to manage and support the financial operations of the organization.

When security appears to trump functionality, it's likely that the real function of the asset is security-related. (military, life-safety, etc)

Functionality *always* trumps security

Incident response takes place in the presence of *high uncertainty*

High Uncertainty means we might not have seen this coming, at least not as a specific event in time. We may not be sure at first what is happening. Things we do with the best intentions might make things worse, or obscure the actual causes.

At the user/organizational level, Functionality is demanded. The security of systems and processes is often merely *assumed*. (“Life is not fair”)

Incident Response can be hard: The kitchen is hot, everything is sharp, and the lights keep going out. (see “Life”, above)

Challenges

People, of course, are the biggest challenge.

Challenges

Fear and Uncertainty
cause delay and *confusion*

People, of course, are the biggest challenge.

Challenges

Fear and Uncertainty
cause delay and *confusion*

Time and Trust Pressures
make achieving consensus
on incident response *hard*

People, of course, are the biggest challenge.

Incident Management

With luck, you have smart security people, and you've done your homework.

Early notice of problems via trusted channels, if you're lucky.

Determine what your exposure is

Figure out what you'll do about it

Communicate to user community, and take steps to prevent problems

Incident Management

Border Protections

With luck, you have smart security people, and you've done your homework.

Early notice of problems via trusted channels, if you're lucky.

Determine what your exposure is

Figure out what you'll do about it

Communicate to user community, and take steps to prevent problems

Incident Management

Host-based protections

With luck, you have smart security people, and you've done your homework.

Early notice of problems via trusted channels, if you're lucky.

Determine what your exposure is

Figure out what you'll do about it

Communicate to user community, and take steps to prevent problems

Incident Management

Policy Implementation
Organizational
Technological

With luck, you have smart security people, and you've done your homework.

Early notice of problems via trusted channels, if you're lucky.

Determine what your exposure is

Figure out what you'll do about it

Communicate to user community, and take steps to prevent problems

Incident Management

User Education and Awareness

With luck, you have smart security people, and you've done your homework.

Early notice of problems via trusted channels, if you're lucky.

Determine what your exposure is

Figure out what you'll do about it

Communicate to user community, and take steps to prevent problems

Getting this right is hard

Security Response is hard, and sometimes, the other kids don’t much like us…

Getting this right is hard

Most actions have side effects

Security Response is hard, and sometimes, the other kids don’t much like us…

Getting this right is hard

Most actions have side effects

Most side effects represent costs

Security Response is hard, and sometimes, the other kids don’t much like us…

Getting this right is hard

Most actions have side effects

Most side effects represent costs

New costs will meet resistance

Security Response is hard, and sometimes, the other kids don’t much like us…

Image stolen under the assumption that “Anarchy, Inc.” probably doesn’t have a lot of intellectual property attorneys.

University networks are rather more subject to individual freedom and choice than in most organizations...

Image stolen under the assumption that “Anarchy, Inc.” probably doesn’t have a lot of intellectual property attorneys.

Patches are hard.

Patches are hard. They must be written under pressure, usually by the group “responsible” for the problem at hand

Patches must be tested thoroughly, because damage to systems, not to mention further damage to vendor reputation, is at stake

> “Pick One...”

Patches are hard.

Timely?

Patches are hard. They must be written under pressure, usually by the group “responsible” for the problem at hand

Patches must be tested thoroughly, because damage to systems, not to mention further damage to vendor reputation, is at stake

> “Pick One...”

Patches are hard.

Timely?

Complete?

Patches are hard. They must be written under pressure, usually by the group “responsible” for the problem at hand

Patches must be tested thoroughly, because damage to systems, not to mention further damage to vendor reputation, is at stake

> “Pick One...”

Patches are hard.

Timely?

Complete?

Safe?

Patches are hard. They must be written under pressure, usually by the group “responsible” for the problem at hand

Patches must be tested thoroughly, because damage to systems, not to mention further damage to vendor reputation, is at stake

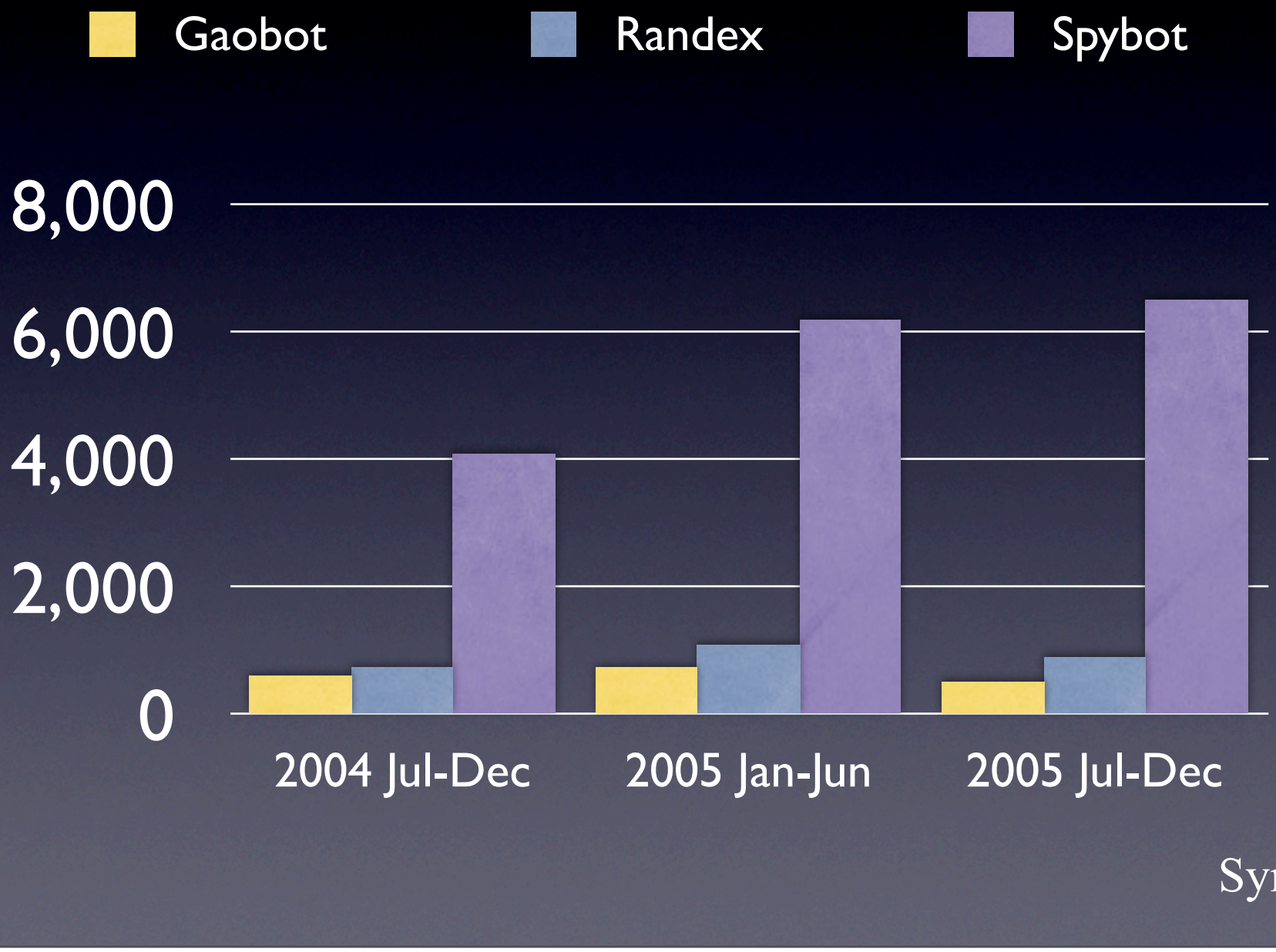
“Pick One...”

Trends

This section on threat “trends” is based on analysis of recent reports and publications done by Dan Geer, who who generously shared his work with me for use in this talk. Sources include the CSI/FBI Report, the Symantec Threat Report, the Anti-Phishing Working Group, Webroot Software, and the Counterpane Attack Trends Report.

My thanks to him for sharing his work and insights, and for taking all those statistics courses... (I wasn’t going to pass anyway)

Top 3 new Bot variants



Note that for Spybot, this is 1.5 new variants Every Hour

Dan: “One can almost consider variation rates like this to be denial of service (DoS) attacks on the computer immune system.”

Data Source: Symantec Threat Report

Phishing

From Anti-Phishing WG data

Phishing

Phishing email reports are up 35%

From Anti-Phishing WG data

Phishing

Phishing email reports are up 35%

Number of URLs used is up 250%

From Anti-Phishing WG data

Phishing & Malware

From Anti-Phishing WG data

Phishing & Malware

Over the past year:

From Anti-Phishing WG data

Phishing & Malware

Over the past year:

172% increase in malware variants

From Anti-Phishing WG data

Phishing & Malware

Over the past year:

172% increase in malware variants

324% increase in urls used

From Anti-Phishing WG data

Phishing & Malware

There are over 200 new variants of ride-along malware each month

From Anti-Phishing WG data

Us *vs.* Them

We (defenders) need to work harder as the number of attacks increases.

They (attackers) need to work only hard enough to make the next variant.

“As [creating new variants] is now automated, the arms race between attacker and defender can be manipulated by the attacker to bankrupt the defender.”

Us *vs.* Them

Malware variants increased by 28X

We (defenders) need to work harder as the number of attacks increases.

They (attackers) need to work only hard enough to make the next variant.

“As [creating new variants] is now automated, the arms race between attacker and defender can be manipulated by the attacker to bankrupt the defender.”

Us *vs.* Them

Malware variants increased by 28X

Phishing urls increased by 35X

We (defenders) need to work harder as the number of attacks increases.

They (attackers) need to work only hard enough to make the next variant.

“As [creating new variants] is now automated, the arms race between attacker and defender can be manipulated by the attacker to bankrupt the defender.”

Us *vs.* Them

Malware variants increased by 28X

Phishing urls increased by 35X

Defender's work factor is cumulative

We (defenders) need to work harder as the number of attacks increases.

They (attackers) need to work only hard enough to make the next variant.

"As [creating new variants] is now automated, the arms race between attacker and defender can be manipulated by the attacker to bankrupt the defender."

Us *vs.* Them

Malware variants increased by 28X

Phishing urls increased by 35X

Defender's work factor is cumulative

Attacker's work factor is the cost of a new variant

We (defenders) need to work harder as the number of attacks increases.

They (attackers) need to work only hard enough to make the next variant.

"As [creating new variants] is now automated, the arms race between attacker and defender can be manipulated by the attacker to bankrupt the defender."

"When you are dealing with rootkits and some advanced spyware programs, the only solution is to rebuild from scratch. In some cases, there really is no way to recover without nuking the systems from orbit."

Mike Danseglio, Program Manager, Security Solutions Group, Microsoft, April 3, 2006.

"Reality Bites"

Microsoft's *Antivirus Defense-in-Depth Guide*

"...viruses, worms, and Trojan horses continue to infect computer systems around the world.

There is no single reason for this apparent contradiction, but the current situation indicates that the standard approach of deploying antivirus software on each computer in your environment may not be sufficient."

http://www.microsoft.com/technet/security/topics/serversecurity/avdind_0.mspx

Zero-Day Project Goals:

Project Summary: “We would like you to dip our product in a variety of poisons and hot oils, and see what happens.”

Zero-Day Project Goals:

Consider Digital Guardian's role in a successful Defense in Depth strategy

Project Summary: "We would like you to dip our product in a variety of poisons and hot oils, and see what happens."

Zero-Day Project Goals:

Consider Digital Guardian's role in a successful Defense in Depth strategy

Investigate DG's ability to provide protection against Zero-Day attacks

Project Summary: "We would like you to dip our product in a variety of poisons and hot oils, and see what happens."

Approach

Simply put, if we can do this, we win.

Approach

Protect User Data

Simply put, if we can do this, we win.

Approach

Protect User Data

Prevent Network Abuse

Simply put, if we can do this, we win.

Approach

Protect User Data

Prevent Network Abuse

Protect Local OS

Simply put, if we can do this, we win.

Approach

Protect User Data

Prevent Network Abuse

Protect Local OS

Protect Local Applications

Simply put, if we can do this, we win.

Zanshin Malware Lab

Zanshin Security assembled a library of malware from existing collections (including the Nepenthes library as of February 2006) and private efforts. At the time of this project, Our malware library included some 3260 unique worm samples (of which 2552 were Korgo/Padobot variants).

We selected 24 worms to serve as a representative sample, covering 93% of our library. Our samples included one of each uniquely-identified variant from each worm type/family. Identification was performed using ClamAV.

Nepenthes has since merged with the mwcollect project: http://nepenthes.mwcollect.org/

Zanshin Malware Lab

Malware Library (*Nepenthes*, plus)

Zanshin Security assembled a library of malware from existing collections (including the Nepenthes library as of February 2006) and private efforts. At the time of this project, Our malware library included some 3260 unique worm samples (of which 2552 were Korgo/Padobot variants).

We selected 24 worms to serve as a representative sample, covering 93% of our library. Our samples included one of each uniquely-identified variant from each worm type/family. Identification was performed using ClamAV.

Nepenthes has since merged with the mwcollect project: http://nepenthes.mwcollect.org/

Zanshin Malware Lab

Malware Library (*Nepenthes*, plus)

VMware on top of Red Hat

Zanshin Security assembled a library of malware from existing collections (including the Nepenthes library as of February 2006) and private efforts. At the time of this project, Our malware library included some 3260 unique worm samples (of which 2552 were Korgo/Padobot variants).

We selected 24 worms to serve as a representative sample, covering 93% of our library. Our samples included one of each uniquely-identified variant from each worm type/family. Identification was performed using ClamAV.

Nepenthes has since merged with the mwcollect project: http://nepenthes.mwcollect.org/

Zanshin Malware Lab

Malware Library (*Nepenthes*, plus)

VMware on top of Red Hat

Physical machines as servers and other infrastructure systems

Zanshin Security assembled a library of malware from existing collections (including the Nepenthes library as of February 2006) and private efforts. At the time of this project, Our malware library included some 3260 unique worm samples (of which 2552 were Korgo/Padobot variants).

We selected 24 worms to serve as a representative sample, covering 93% of our library. Our samples included one of each uniquely-identified variant from each worm type/family. Identification was performed using ClamAV.

Nepenthes has since merged with the mwcollect project: http://nepenthes.mwcollect.org/

Zanshin Malware Lab

Malware Library (*Nepenthes*, plus)

VMware on top of Red Hat

Physical machines as servers and other infrastructure systems

Strictly "air-gapped" network

Zanshin Security assembled a library of malware from existing collections (including the Nepenthes library as of February 2006) and private efforts. At the time of this project, Our malware library included some 3260 unique worm samples (of which 2552 were Korgo/Padobot variants).

We selected 24 worms to serve as a representative sample, covering 93% of our library. Our samples included one of each uniquely-identified variant from each worm type/family. Identification was performed using ClamAV.

Nepenthes has since merged with the mwcollect project: http://nepenthes.mwcollect.org/

Malware Injection

http://www.metasploit.com/

Malware Injection

Metasploit framework

http://www.metasploit.com/

Malware Injection

Metasploit framework

Servers in the test environment

http://www.metasploit.com/

Malware Injection

Metasploit framework

Servers in the test environment

Manually

http://www.metasploit.com/

Malware
Strategy & Tactics

Malware
Strategy & Tactics

Infection mechanisms and targets

Malware
Strategy & Tactics

Infection mechanisms and targets

Propagation

Malware
Strategy & Tactics

Infection mechanisms and targets

Propagation

Self-Preservation

Targets

Targets

Executable files

Targets

Executable files

Documents and data files

Propagation

Propagation

Removable storage

Propagation

Removable storage

Email and other network downloads

Self-preservation

Stealth by Design Malware doesn’t rely on conventional rootkit technology to hide itself, instead makes stealth a core design goal.

See http://invisiblethings.org

Polymorphic code is code that mutates while keeping the original algorithm intact.

Metamorphic code is code that can reprogram itself. Often, it does this by translating its own code into a temporary representation, and then back to normal code again. This is used by some viruses when they are about to infect new files, and the result is that their "children" will never look like themselves.

(see http://en.wikipedia.org/ and many other sources)

Self-preservation

Stealth by Design *(SbD)*

Stealth by Design Malware doesn’t rely on conventional rootkit technology to hide itself, instead makes stealth a core design goal.

See http://invisiblethings.org

Polymorphic code is code that mutates while keeping the original algorithm intact.

Metamorphic code is code that can reprogram itself. Often, it does this by translating its own code into a temporary representation, and then back to normal code again. This is used by some viruses when they are about to infect new files, and the result is that their "children" will never look like themselves.

(see http://en.wikipedia.org/ and many other sources)

Self-preservation

Stealth by Design *(SbD)*

Polymorphism and metamorphism

Stealth by Design Malware doesn’t rely on conventional rootkit technology to hide itself, instead makes stealth a core design goal.

See http://invisiblethings.org

Polymorphic code is code that mutates while keeping the original algorithm intact.

Metamorphic code is code that can reprogram itself. Often, it does this by translating its own code into a temporary representation, and then back to normal code again. This is used by some viruses when they are about to infect new files, and the result is that their "children" will never look like themselves.

(see http://en.wikipedia.org/ and many other sources)

Self-preservation

Stealth by Design *(SbD)*

Polymorphism and metamorphism

Antivirus deactivation

Stealth by Design Malware doesn't rely on conventional rootkit technology to hide itself, instead makes stealth a core design goal.

See http://invisiblethings.org

Polymorphic code is code that mutates while keeping the original algorithm intact.

Metamorphic code is code that can reprogram itself. Often, it does this by translating its own code into a temporary representation, and then back to normal code again. This is used by some viruses when they are about to infect new files, and the result is that their "children" will never look like themselves.

(see http://en.wikipedia.org/ and many other sources)

Malware very often uses predictable pathnames that have little or no overlap with pathnames important to human users wishing to create files.

Malware propagation mechanisms often use network ports that are needed by relatively few, well-known, legitimate system programs.

Although malware does often write to the registry, it is uncommon for registry changes alone to be sufficient for the malware's viability, and it is rare for the registry changes to have an independent impact on the integrity or usability of the system.

Malware often uses predictable pathnames

Malware very often uses predictable pathnames that have little or no overlap with pathnames important to human users wishing to create files.

Malware propagation mechanisms often use network ports that are needed by relatively few, well-known, legitimate system programs.

Although malware does often write to the registry, it is uncommon for registry changes alone to be sufficient for the malware's viability, and it is rare for the registry changes to have an independent impact on the integrity or usability of the system.

Malware often uses predictable pathnames

Malware propagation often uses network ports that are needed by a small number of known, legitimate programs

Malware very often uses predictable pathnames that have little or no overlap with pathnames important to human users wishing to create files.

Malware propagation mechanisms often use network ports that are needed by relatively few, well-known, legitimate system programs.

Although malware does often write to the registry, it is uncommon for registry changes alone to be sufficient for the malware's viability, and it is rare for the registry changes to have an independent impact on the integrity or usability of the system.

Malware often uses predictable pathnames

Malware propagation often uses network ports that are needed by a small number of known, legitimate programs

It’s uncommon for registry changes alone to be sufficient for the malware's viability

Malware very often uses predictable pathnames that have little or no overlap with pathnames important to human users wishing to create files.

Malware propagation mechanisms often use network ports that are needed by relatively few, well-known, legitimate system programs.

Although malware does often write to the registry, it is uncommon for registry changes alone to be sufficient for the malware's viability, and it is rare for the registry changes to have an independent impact on the integrity or usability of the system.

Initial Results

In the actual tests performed, we found that 30% of network worms were completely blocked, and the remaining 70% were ineffective but left some artifacts on the system.

In one of these cases, the vulnerability was not exploited successfully, causing LSASS.EXE to crash and the system to reboot 60 seconds later (the expected result under these conditions).

Initial Results

30% of worms tested were blocked outright

In the actual tests performed, we found that 30% of network worms were completely blocked, and the remaining 70% were ineffective but left some artifacts on the system.

In one of these cases, the vulnerability was not exploited successfully, causing LSASS.EXE to crash and the system to reboot 60 seconds later (the expected result under these conditions).

Initial Results

30% of worms tested were blocked outright

The remaining 70% of attacks were rendered ineffective.

In the actual tests performed, we found that 30% of network worms were completely blocked, and the remaining 70% were ineffective but left some artifacts on the system.

In one of these cases, the vulnerability was not exploited successfully, causing LSASS.EXE to crash and the system to reboot 60 seconds later (the expected result under these conditions).

Initial Results

30% of worms tested were blocked outright

The remaining 70% of attacks were rendered ineffective.

We identified a possible extension to the ruleset language which will cover 100% of the worm, virus, and Trojan horse attack vectors.

In the actual tests performed, we found that 30% of network worms were completely blocked, and the remaining 70% were ineffective but left some artifacts on the system.

In one of these cases, the vulnerability was not exploited successfully, causing LSASS.EXE to crash and the system to reboot 60 seconds later (the expected result under these conditions).

Initial Results

30% of worms tested were blocked outright

The remaining 70% of attacks were rendered ineffective.

We identified a possible extension to the ruleset language which will cover 100% of the worm, virus, and Trojan horse attack vectors.

In the actual tests performed, we found that 30% of network worms were completely blocked, and the remaining 70% were ineffective but left some artifacts on the system.

In one of these cases, the vulnerability was not exploited successfully, causing LSASS.EXE to crash and the system to reboot 60 seconds later (the expected result under these conditions).

Digital Guardian & Incident Response

The background for the sort of IR events we're considering was discussed in more detail in our paper on incident response and large event management:

http://www.zanshinsecurity.com/archive/zanshin-incidentresponse.pdf

Digital Guardian & Incident Response

We can push out rules to block a specific activity

The background for the sort of IR events we're considering was discussed in more detail in our paper on incident response and large event management:

http://www.zanshinsecurity.com/archive/zanshin-incidentresponse.pdf

Digital Guardian & Incident Response

We can push out rules to block a specific activity

We have the agility to rapidly refine the rule as new information warrants

The background for the sort of IR events we're considering was discussed in more detail in our paper on incident response and large event management:

http://www.zanshinsecurity.com/archive/zanshin-incidentresponse.pdf

Digital Guardian & Incident Response

We can push out rules to block a specific activity

We have the agility to rapidly refine the rule as new information warrants

This capability can be built into security policies and procedures ahead of time

The background for the sort of IR events we're considering was discussed in more detail in our paper on incident response and large event management:

http://www.zanshinsecurity.com/archive/zanshin-incidentresponse.pdf

Traditional Approaches

Traditional Approaches

Emergency update of virus definitions?

Traditional Approaches

- Emergency update of virus definitions?
  - Might not be effective, and malware might disable antivirus

Traditional Approaches

- Emergency update of virus definitions?
  - Might not be effective, and malware might disable antivirus
- Network blocking?

Traditional Approaches

- Emergency update of virus definitions?
  - Might not be effective, and malware might disable antivirus
- Network blocking?
  - Not all hardware, not all network topologies

Traditional Approaches

- Emergency update of virus definitions?
  - Might not be effective, and malware might disable antivirus
- Network blocking?
  - Not all hardware, not all network topologies
- Patch and reboot every machine?

Traditional Approaches

Emergency update of virus definitions?

- Might not be effective, and malware might disable antivirus

Network blocking?

- Not all hardware, not all network topologies

Patch and reboot every machine?

Labor-intensive and time-consuming, not possible in the case of 0day events

How can Digital Guardian help?

How can Digital Guardian help?

The network traffic required to deploy a new rule is a fraction of that required by a patch

How can Digital Guardian help?

The network traffic required to deploy a new rule is a fraction of that required by a patch

Machines can be updated without requiring a reboot, unlike most patches

How can Digital Guardian help?

The network traffic required to deploy a new rule is a fraction of that required by a patch

Machines can be updated without requiring a reboot, unlike most patches

All of this is done on the organization's schedule, and focused on their priorities

Example DG Response

Example DG Response

IT Staff sees increased port 135 traffic

Example DG Response

IT Staff sees increased port 135 traffic

Action: Deploy ruleset blocking port 135

Example DG Response

IT Staff sees increased port 135 traffic

Action: Deploy ruleset blocking port 135

Result?

Example DG Response

IT Staff sees increased port 135 traffic

Action: Deploy ruleset blocking port 135

Result?

No new infections

Example DG Response

IT Staff sees increased port 135 traffic

Action: Deploy ruleset blocking port 135

Result?

No new infections

Network utilization returns to normal

Example DG Response

IT Staff sees increased port 135 traffic

Action: Deploy ruleset blocking port 135

Result?

No new infections

Network utilization returns to normal

We've bought ourselves analysis time

Response Continues

Response Continues

Tentative conclusions:

Response Continues

Tentative conclusions:

- Port 135 traffic was a worm

Response Continues

Tentative conclusions:

- Port 135 traffic was a worm
- Some port 135 traffic is important

Response Continues

Tentative conclusions:

- Port 135 traffic was a worm
- Some port 135 traffic is important

Action:

Response Continues

Tentative conclusions:

- Port 135 traffic was a worm
- Some port 135 traffic is important

Action:

- *Protect servers & Domain Controllers*

Response Continues

Tentative conclusions:

- Port 135 traffic was a worm
- Some port 135 traffic is important

Action:

- *Protect servers & Domain Controllers*
- *Refine ruleset to include exceptions for Server and Domain Controller addresses*

Analysis?

Analysis?

We have stopped spread of the worm

Analysis?

We have stopped spread of the worm

We now have time to patch and clean up

Analysis?

We have stopped spread of the worm

We now have time to patch and clean up

We had functional blocking without resorting to infrastructure blocking

Analysis?

We have stopped spread of the worm

We now have time to patch and clean up

We had functional blocking without resorting to infrastructure blocking

We served local needs and priorities, with much better control over schedule

Analysis?

We have stopped spread of the worm

We now have time to patch and clean up

We had functional blocking without resorting to infrastructure blocking

We served local needs and priorities, with much better control over schedule

What if it had been a port 80 worm?

Main points

Main points

- Digital Guardian offers a powerful and flexible new tool against observed or predictable malicious activity

Main points

- Digital Guardian offers a powerful and flexible new tool against observed or predictable malicious activity
- Digital Guardian presents a very attractive ad-hoc response capability in emergent situations

Main points

- Digital Guardian offers a powerful and flexible new tool against observed or predictable malicious activity
- Digital Guardian presents a very attractive ad-hoc response capability in emergent situations
- This capability empowers organizations to respond to threats effectively, and with local priorities in mind

Q?

Bob Mahoney, Principal • Zanshin Security, LLC • http://zanshinsecurity.com

Very Special thanks to our malware project staff, Alejandro Sedeno and Matt Power. You folks do *cool* work...

Some Other References

The Metasploit Project
OSVDB: The Open Source Vulnerability Database
CVE - Common Vulnerabilities and Exposures
Common Malware Enumeration (CME)
mwcollect.org
Mal-Aware.org

Q?

"Hey, hey, hey- Don't be mean."

-Buckaroo Banzai

bob@zanshinsecurity.com

Bob Mahoney, Principal • Zanshin Security, LLC • http://zanshinsecurity.com

Very Special thanks to our malware project staff, Alejandro Sedeno and Matt Power. You folks do *cool* work...

Some Other References

The Metasploit Project
OSVDB: The Open Source Vulnerability Database
CVE - Common Vulnerabilities and Exposures
Common Malware Enumeration (CME)
mwcollect.org
Mal-Aware.org