RAPIER: A 1st Responders Information Acquisition Framework

Abstract: RAPIER (Rapid Assessment & Potential Incident Examination Report) is a security tool built to facilitate 1st response procedures for incident handling. It is designed to acquire commonly requested information and samples during an information security event, incident, or investigation. RAPIER automates the entire process of data collection and delivers the results directly to the hands of a skilled security analyst.

Steve Mancini and Joseph Schwendt

Executive Summary

For most organizations the first significant obstacle that confronts their incident handling process is the ability to full comprehend the attack that is underway. Waiting for vendor protection services to provide answers can exact an extreme cost for many companies where degraded performance or complete outages equates to financial loss. It is essential for an organization to empower its response team with the tools necessary to quickly obtain the information required for security analysts to determine the characteristics of the current incident so that they can either identify it as a known vector, or report it to the appropriate sources to seek remediation for the new attack. RAPIER is an open source modular framework designed to allow 1st responders to execute information gathering scans on a Microsoft Windows system which capture specified data both on the system and in volatile memory. Through automation security analyst resources are freed up to focus on the analysis of data which is acquired in a deterministic manner.

It is essential for an organization to empower its response team with the tools necessary to quickly obtain the information required for security analysts…

Problem Statement

Over the last several years there has been an increasing demand for efficiency, agility, and speed in the incident handling discipline. This demand is further challenged when the target organization is globally dispersed or operates with extended business hours that overflow from the traditional "business day" requiring an on-call strategy to provide response coverage1. As with any security solution, there is also the concern for costs involved in providing the necessary and sufficient resources to response to an incident. Providing a globally responsive team of skilled responders incurs a significant cost in personnel, tools, and training.

Depending on the organization and the funding they wish to assign to their solution, establishing such a response program can be challenging. Procedures that rely upon the experience and expertise of the incident handler to be 1st responders who gather information can introduce variance in the process. Different incident handlers may select different tools to accomplish the same task (how many tools can do a port scan these days?), they may rely on different versions of the tools, they may handle the data they retrieve differently, etc. While a complete investigation will often introduce a point after which "out of the box" thinking is required, the first stages of handling an incident bears the potential to be handled in a deterministic manner.

The problem for most companies is achieving this copy exact methodology both in the tool suites and processes used at a reasonable cost. For many organizations with a dispersed presence, this may translate into experts walking extended members of staff, or worse the end-user, through the regiments of acquiring data. When the attack possesses a volatility that makes it too dangerous to allow the system to remain on the network, the acquisition and transferal of information is further aggravated. For many organizations, "pull the power plug" is still considered the best solution even though potentially critical data is lost by this archaic BKM.

Capturing data that is necessary and sufficient to facilitate the analysts who would seek to determine the events leading up to the current incident generates an extensive list of requirements. For organizations where uptime is a top priority, the security analyst needs to acquire all data they need from the start before the system is put through their regimented programs of recovery. Often it is critical that evidence acquisition occur *before any modifications to the system take place.* This data is not always static on the system – with the evolution of malware there are numerous system characteristics that must be acquired before the system is modified. Below is the beginning of a list of potentially critical information to obtain for the purpose of event analysis.

Volatile Information	Static Information
List of running processes	System Name
Locations of processes on disk	System Startup Commands
Ports those processes are using	Copies of application cache (temporary internet files)
Net start/share/user/file/session)	Uptime
Layer3 traffic samples	

[1] It's 4am. You are asleep. Then the phone rings and a frantic member of the IT organization who is wide awake during their normal shift begins rattling off information at you expecting a coherent strategy right now!

Output from nbtstat and netstat

Dump memory for all running processes

Checksums for all running processes

Status of NIC(s) – promiscuous mode?

Capture last Modify/Access/Create times for designated areas of the target system

Document all open shares/exports on system

All files that are currently open

Capture current routing tables

All DLLS currently loaded and their checksum

capture logged in users

list of all network connections

Local account and policy information

List of all files with alternate data streams

Capture list of services installed on the system

Discover files marked as hidden

Export entire registry

Current patches installed on system

Current AV versions

list of all installed software on system

Capture all logs (system + application specific)

MAC address

Solution

From the outset we must acknowledge that this idea is not new. Inspiration was derived from a presentation made by Jesse Kornblum on his work producing a first responders program (FRED[2]) for earlier versions of the Windows operating system. Jesse's program was designed to operate on a floppy to capture some basic information from the system. Our approach proceeds along the same lines but with greater functionality, innovation, and with a focus on current versions of the Windows operating system. This solution also was designed to meet the needs of analysts – capturing the information that was considered crucial to the execution of their role in an event or incident.

The incident handling discipline strives for efficiency and speed by adopting many of the strategies for crisis response from other disciplines. Proven methodologies from areas such as the military, civilian response services, and emergency room procedures all provide an excellent foundation from which a successful incident handling process can be created. What these industry best practices teach us is that to be efficient and agile requires a process that:

- Introduces a limited number of decisions by the 1st responder that could result in differing results
- Automates where possible to free up the incident handler's focus on issues of greater concern
- Relies upon deterministic tools that can function with equal effect on like systems
- Provides for a complete lifecycle for information gathering from start to delivery of data

[2] http://www.csa.syr.edu/Jesse_Kornblum.pdf, http://research.jessekornblum.com/presentations/simple-but-sound.pdf

- Expedites the acquisition of information since time is of the essence
- Provides the ability to expand functionality quickly to adapt to immediate needs in a short time frame
- Comprehends data that could be requested by analysts and gathers it during 1st execution of the tool

The conceptual architecture behind RAPIER incorporated these best practices to provide a framework through which the information that is sought by incident response analysts could be obtained. RAPIER is essentially a framework (engine) which runs individual modules to collect information and upload it to the central secured repository where analysts can then examine the output from the program. It is designed to be run locally on suspect machines in unaltered state. Much like the original FRED disk, the program is able to be run from a removable media, such as a USB drive. RAPIER has proven to be extremely effective in helping to understand the impact of malware on systems where it can highlight the changes made to a system as well as potentially capture samples of the added code. Serendipitous benefits include the ability for system administrators to use RAPIER as a debugging tool since it provides a great deal of information about the computer it is run on.

Features

- Modular Design with dynamic adopting of new modules – Allows for rapid development of new information gathering methods. Can be custom tailored to any environment.
- Fully configurable GUI (Graphical User Interface) – Standard Windows application look and feel. Easy to use.
- Auto-update functionality with SHA1 verification – Entire application is self updating. Checks for new version on startup (if connected to network). Uses SHA1 checksums to ensure files are verifiably current and authentic. Only those files in need of updating are transferred, which ensures network bandwidth is conserved.
- Results are auto-zipped – Typical compression of results is nearly 10x, thus reducing transfer times. The zip file is also password protected with a unique password to ensure virus scanners do not interfere with transferal and storage of results.
- Results are auto-uploaded to central secured repository – If system is online, results are automatically sent to a central repository where a security analyst can analyze them in a lab environment.
- Email notifications –the program can be configured to send an email upon completion to recipients designated in the configuration files. This has proven to be a useful method for alerting analysts that there is an issue requiring their attention
- Separate Output – the output from each module is stored in a separate file allowing for easier handling as well as the ability to route the information into automated analysis programs off system.
- Pre/Post Run Integrity Check – the system is examined before and after the execution of RAPIER to allow for the documentation of all modifications performed by the tool to the system.

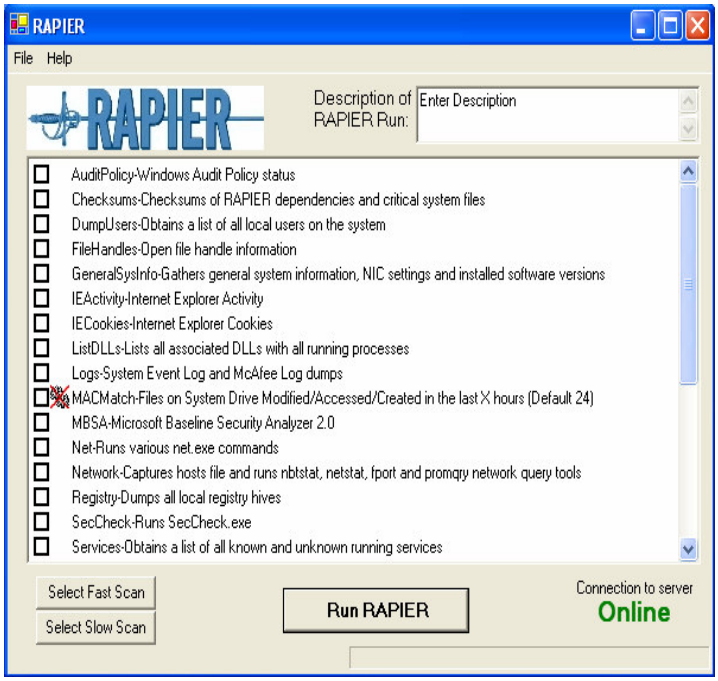
- Command Line Configuration/Execution – RAPIER has been designed to allow for execution from the command line obtaining its parameters either from the command line, or through the configuration files.

Requirements

- Windows NT based Operating System (Windows 2000, Windows XP, Windows 2003, Windows Vista)
- Microsoft .NET Framework 1.1 or greater (included in Windows 2003 and newer)
- Microsoft WSH (Windows Scripting Host) 5.6 or greater (included in Windows 2000 SP3+ as well as Windows XP and newer)
- Microsoft WMI (Windows Management Interface) 1.5 or greater (included in Windows 2000 SP3+ as well as Windows XP and newer)

RAPIER in Action

RAPIER is designed with ease of use in mind. It is expected that any general support organization can run it either under the guidance of security analysts or through the use of default instructions defined as part of an organizations response program. RAPIER can be made available internally as a ZIP file via an http website. The 1st responder or end-user is able to download the engine, or engine and modules[3], to their local machine or to the external writable media of choice (thumb drive, USB hard drive, etc). They should then run RAPIER.exe by double clicking on it. If online, RAPIER will automatically check to ensure it is the latest version. The application features the ability to validate the integrity of the entire program itself from a secure source (SHA1 checksum verified). RAPIER does not extend its footprint beyond the directory it is launched from unless otherwise specified in the options screen.



Once the graphical user interface is loaded, the user selects the appropriate modules for the given situation. This step is often pre-established by an organization's rules of engagement otherwise the modules to be selected should be directed by the analyst receiving the results. Also included are two default options; users can select "Fast Scan" which enables the most common information to be gathered, and will take approximately 10 minutes to run. After selecting the appropriate modules, the user simply needs to click on the Run RAPIER button. If the

[3] The engine only option is included to remove the redundancy of downloading the modules twice – during the initial download and again later if/when they execute an update.

system is online when running RAPIER, the results will be automatically uploaded at the end of running the selected modules, otherwise you will have the locally (relative to where ever you installed RAPIER) stored copy.

RAPIER is designed to collect volatile state information from the target system. For this reason the end user should not disconnect, shutdown, or alter the system state until after running RAPIER unless directed to do so by the security analyst. This may alter the effectiveness of collecting malware samples since some malware is now smart enough to detect when the network has been disconnected and go into hibernation. If however the system is causing dramatic harm to the environment, disconnecting that system prior to running RAPIER is a good idea.

Under the Hood: Engine and Module designs

Engine Design

RAPIER 3.0 is essentially a framework (engine) which runs individual modules to collect information and upload it to the central repository. The RAPIER engine is written in Microsoft Visual Basic .NET 2003, and thus requires the Microsoft .NET Framework 1.1 be installed on the target system. It also utilizes several external tools in order to perform online connectivity and ZIP compression.

The engine is responsible for the following tasks

- Determine if the system is online and can contact the server
- Determine if the application and modules are up to date and verifiably authentic (if online)
- Update the application and/or modules
- Present the user with a GUI (Graphical User Interface)
- Allow the user to configure the following options:
    - Module Directory – Directory where the RAPIER modules are located. Defaults to a directory called "Modules" in the same directory as the RAPIER.exe
    - Results Directory – Directory where the results from running the selected RAPIER modules should be placed. Defaults to a directory called "Results" in the same directory as the RAPIER.exe
- Auto-ZIP results – Enables/Disables the automatic ZIPing (compression) of the results at the end of running the selected modules.
- Auto-Upload results – Enables/Disables the automatic uploading of the results at the end of running the selected modules. Only enable-able if the system is online and Auto-ZIP results are enabled.
- Allow the user to upload a results ZIP file collected from an offline system
- Allow the users to view online help
- Allow the user to individually select the modules or select the Fast or Slow scan
- Runs the selected modules. Passes the path to the results directory to each individual module

Module Design

RAPIER is designed not to re-invent the wheel. Several talented software designers have provided numerous tools that enable the acquisition of the information most security analyst require during an investigation. Each of the tools selected for inclusion with RAPIER was chosen because it either provided information as a first source, or in some cases, was a redundant verification of the system state. Part of the approach taken with RAPIER is that using different tools to acquire the same information is not a bad thing – if there are unexplainable differences in the output this may indicate there is more going on than initially anticipated.

RAPIER module wrappers are written in Microsoft VBScript. They require Microsoft WSH (Windows Scripting Host) 5.6 and Microsoft WMI (Windows Management Interface) 1.5 to be installed on the target system. They also utilize several VBScript libraries and external tools in the "Tools" directory. Included among the modules are three designed by the authors or their peers to meet in-house needs. RAPIER.vbi is an extensive library developed as part of system administration and patching processes. Some of the external tools have been developed in-house explicitly for use with RAPIER, but could potentially be used by any application. These include AutoproxyResolver.exe (Resolves the proxy server for a given target URL based on a given Autoproxy URL) and L3Sniffer.exe (Layer 3 (IP stack) network sniffer which dumps raw packets in byte format that can be easily converted to tcpdump format with the use of text2pcap.exe from Ethereal). Individual module behavior can also be controlled though configuration files in the "Conf" directory located in the same directory as RAPIER.exe. Modules will need to be written to take specific advantage of this functionality.

Using the VBScript libraries, module development is a very rapid process. A new module can be developed and tested within an hour to enable rapid response to a new type of malware. The engine's self updating mechanism ensures that new modules can be deployed as rapidly as they are available. If a module is no longer needed, the auto-updating mechanism will ensure it is removed from the system.

Each module resides in its own directory which is the name that will appear in the RAPIER GUI. These folders are located in one of three folders:

- Fast – for modules which will execute quickly and allow for the Fast Scan to remain under 10 minutes (in most cases)
- Slow – for the modules which take longer to execute
- Special – for extremely invasive scans and examinations which can potentially take a great deal of time and thus has no assurance of completing within a relatively predictable amount of time.

The folder will contain 3 or more files. A `Module.cmd` file is used to call the executables via the `Module.wsf` wrapper that is used to control the executation and output of the included binary. Additional files, such as .ini files or other configuration files, are included on a "as-needed" basis. The wrapper is built to accept one command line option, which is the path to the directory to place the module results. Each module should create a <Module Name>.log file in the results directory, which contains basic information. Some modules may need to create additional files or directories in the results directory. For example, the DumpProcs module creates a directory and copies the exact executable from memory of each process that is currently running on the system to that directory.

Feature Modules

RAPIER comes bundled with a rich set of modules. While this list contains only those modules which we are allowed to disclose to the public, you can see that it is by no means exhaustive. One of our hopes by releasing RAPIER under an open source license is to motivate others to contribute modules and thereby expand the capabilities of the tool and those who use it. Where possible, RAPIER does select tools which are released under licenses which promote sharing and re-packaging. To those individuals and companies the authors of RAPIER wish to express their gratitude.

The modules featured below are from a variety of sources. First preference is given to open source tools which can be re-bundled with RAPIER itself. Others are components which are offered by Microsoft as part of the OS or in their various resource kits. Still other tools are freeware from companies who would not allow the inclusion of their tools with our release bundle. In each of these latter cases the local group supporting the use of RAPIER should legally acquire the necessary binaries to complete their RAPIER modules.

Fast Modules

We define fast modules as those which will typically run within a 5 minute time window. Our end goal is to keep the total execution time for a Fast Scan to around 10 minutes total.

Module	Description:
AuditPolicy:	Reports the status of the built in Windows Audit Policy. If an audit policy is disabled, Windows will not capture information for that type of policy. Several types of malware have been known to disable auditing to cover its tracks.
Checksums	SHA1 Checksums of RAPIER dependencies and critical system files. This is useful for determining if RAPIER and the Operating System are able to function in an uncompromised manor. If any of these files are suspect, so are the results that RAPIER produces.
DumpUsers	Obtains a list of all local users on the system. Non-standard users can typically indicate a trojan or backdoor.
FileHandles	Enumerates a list of all open file handles on the system. Open file handles indicate that a file is being accessed either for reading or writing.
GeneralSysInfo	Gathers a bevy of general system information, NIC (Network Interface Card) settings, and installed software versions. This module provides overview information, which can help to narrow down infection vectors.
IEActivity	Gathers Microsoft Internet Explorer activity from the standard installation location. With the variety of attack vectors based upon sending the victim to a controlled website, this information has proven useful when looking

	for commonly visited sites between users.
IECookies	Collects all stored cookies from Microsoft Internet Explorer. We acquire this information in case we suspect Session ID spoofing may be related to the event under investigation
ListDLLS	Lists all associated DLLs with all running processes. This is extremely useful when looking for extra dlls hooked into processes that are not traditionally associated with the execution of the program.
Logs	Dumps the System, Security, and Application Event Logs, as well as the McAfee Event log.
MACmatch	Lists all files which have been Created, Accessed, and Modified on the system drive within the last 24 hours. This is usually a good starting point to narrow down suspect files.
Net	Runs "net.exe SHARE" (enumerates windows file and printer shares on the system), "net.exe START" (enumerates all running Services), "net.exe USER" (lists all local user accounts), "net.exe USE" (enumerates all mapped windows file and printer shares), "net.exe FILE" (lists open files on the system, the user who has the file open, and the number of locks on the file), and "net.exe SESSION" (lists sessions between the system and other systems).
Network	Runs nbtstat (Displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP)), netstat (Displays protocol statistics and current TCP/IP network connections), fport (Lists all processes and the associated TCP/IP ports) and promqry (Determines if any network interfaces are running in promiscuous mode).
Registry	Dumps all local Registry Hives. Malware often will insert or modify registry keys.
SecCheck	Runs SecCheck.exe, a terrific tool that is a parallel effort along the same lines as RAPIER. This tool provide information about system startup, system, TCP and UDP connection tables, process list, services running on the machine, drivers running, browser helper objects (BHO), etc. This is a slick tool!
Services	Obtains a list of all known and unknown services and their state. Uses the configuration files to determine if services are known.
WinAudit	Runs WinAudit.exe, which is similar to SecCheck, but provides other wide ranging information. It provides a great deal of information about the system itself, its hardware, etc.

Slow Modules

Slow scans are those that take longer to run than the 5 minute window. Often these are tools that execute comprehensive scans of the files on the system, the hard disk, or capture information that requires an extended amount of time to obtain.

Module	Description
ADS	Scans the System Drive for NTFS Alternate Data Streams. A relatively unknown compatibility feature of NTFS, Alternate Data Streams (ADS) provides a method of hiding root kits or hacker tools on a system and allows them to be executed without being detected.
ddPhysMem	Dumps the entire contents of physical memory to a file. Note that the file this creates will be equal in size to the amount of physical memory in a system.
DumpProcs	Dumps the executables of all running processes. This is the most effective means of capturing a malware sample.
HiddenFiles	Lists all hidden files on the system drive and lists their last access time
WebCache	Dumps the Internet Explorer cache. This is also an effective means of capturing a malware sample, since Internet Explorer is a popular infection vector

Special Modules

Special modules are those which have demonstrated extended time frames for execution which are often based upon relatively intrusive actions that look at the content of files or packets. For this reason these modules are infrequently executed during a 1st response procedure but are included for the sake of completeness.

Module	Description
AVScan	Antivirus Command Line Scan of all fixed disks. The module will do an auto-update to the latest DAT file if the system is online prior to performing the scan. It will not clean or delete any files, only scan. This is useful for determining if the suspect malware is already known and detected by the latest DAT file.
FileCapture	A grep like functionality to allow for searching of the system for specific files or files matching a wildcard pattern. This is useful when you suspect a specific type of malware which is know to leave files installed matching a certain naming scheme.
L3Sniffer	Layer 3 (IP stack) network sniffer which runs for 10 minutes or 10,000 packets, which ever comes first. This module produces a tcpdump compatible capture file by using text2pcap.exe from Ethereal to inject dummy Layer 2 information. The capture file can be analyzed using Ethereal or any industry standard packet analyzer.

Results Analysis

The results that RAPIER produces are not designed to be interpreted by the average user. They are tailored towards a trained security analyst, who is knowledgeable in the dark arts of incident handling. While there are many tools emerging in the market that can highlight the abnormalities that are known for static system information, it was recognized early on that such tools are only as valuable as the latest known documented issues. To resist the temptation of growing dependent on others for analysis of the situation, RAPIER will continue to avoid providing any automatic diagnostic analysis of the results it provides. Instead of presuming to tell you what is wrong, our goal is to provide you a complete picture of the machine as it was discovered without requiring the security analyst to be the first responder. Together, the results from the various modules can paint a very detailed picture of the state on the potentially infected system. In the hands of a trained security analyst, this information can be invaluable information.

Future Development

The public release of RAPIER is the 3.0 version. We recognize there is still a great deal of improvement that can be had in both the engine and the modules. Part of our hope is that by releasing this tool to the security community we can interest others in contributing. Some of the areas that we are personally interested in developing, perhaps even after this paper is submitted and before the conference, include:

- Spyware / adware scanner
- Greater scrutiny of device drivers
- Linkage to a known good database for system binary checksums
- Developing Scan options based upon content requests from the AV vendors so that you can provide samples directly to the AV vendors (and freeware counterparts).
- Research using steganography detection tools in Special Scan section.
- Research rootkit revealing technologies for incorporation

Work is underway for a Linux variant of the tool. This effort is being led by Jeff Boerio. For those interested in more information, Jeff can be reached at jeff.boerio<at-sign>intel.com.

Attribution and Gratitude

The authors of RAPIER would like to thank the following individuals for the support and contributions to the production of this tool:

Module Designers and Q&A: Robbie Bytheway, Dan Codorean, Tom Gibb, and Toby Kohlenberg

Logo Design: Amber Bytheway

We would also like to thank the following tool authors who were kind enough to allow us to bundle their tools with the RAPIER release:

Lawrence Baldwin (SecCheck)

http://www.mynetwatchman.com

Jem Berkes (md5sums)

http://www.pc-tools.net

Frank Heynes (LADS tool)

http://www.heysoft.de

Nir Sofer (cprocess)

http://www.nirsoft.net

Arne Vidstrom (macmatch, pmdump)

http://ntsecurity.nu

Kevin Stanush (dumpsec)

http://www.systemtools.com

Appendix 1: ProDiscover

When writing this paper, we noticed the announcements about Techpathway's product, ProDiscover*, and were immediately asked by peers about this product and how it compares to RAPIER. At this point we have not had time to do a side by side comparison. There are always pro's and con's to any selection made including between open source and a vendor product. A cursory glance of ProDiscover indicates the tool has promise – they follow many of the same industry Best Practices as RAPIER. Perhaps in the future we will obtain a copy of the tool and do a comparison of the features on the sourceforge website, but for now we will dodge the issue and allow you to decide which tool meets all of your requirements.

For complete article see: http://www.techpathways.com/ProDiscoverIR.htm

About the Authors

Steve Mancini has been with Intel since May 1997 when he graduated from the Purdue University computer science program. After surviving a year in a technical support role he moved on to UNIX applications where he was a member of the team responsible for building an extensive UNIX application tool suite critical to chip design. In early 2000 he seized the opportunity to pursue his college interest as a security program manager and has since worked as a senior information security specialist and now security strategist. During his time he been involved with several Intel security initiatives including the formation of the Security Operations Center, co-authored of Intel's risk assessment process, and with his interest in incident handling which resulted in his creation of the first generations of RAPIER. Steve has received 3 SANS certifications with honors in Incident Handling and Auditing. In his spare time Steve volunteers as a digital forensics examiner for the city and county police department. For fun he participates in the Defcon Capture the Flag competition. Contact Email: steve.mancini<AT-SIGN>gmail.com

Joe Schwendt joined Intel in January of 2000 with IOS (Intel Online Services). As a Senior Platform Engineer, he helped to advance the design of the Windows build and various other support infrastructures. He also pioneered the first cross platform security monitoring tool in use at Intel. Joe joined IT late in 2001 in Hudson with Engineering Computing. Joe co-developed IPACE (Intel Patch Assistant and Compliance Enabler) and led the EC Windows STET (Security Technologies Engineering Team) for nearly two years. Joe then joined the RRM (Response and Recovery Management) Team in early 2005 as an ITERP (Information Technology Emergency Response Process) Incident Commander and security tools developer. He is the co-developer of RAPIER as well as the Malware Collection Tools Product Manager. Prior to joining Intel, Joe worked for Booz-Allen & Hamilton as a Senior Network Engineer, designing and deploying a border control system for the US Government. Contact Email: rapier<AT-SIGN>schwendt.com