MIRV: A volatile agent to repel intruders

Summary

MIRV (Metasploit's Incident Response Vehicle) is a new tool (based on Metasploit's meterpreter) which was created to address the perceived shortcomings in existing host-based incident response tools: they do not operate on large amounts of nodes, are difficult to get past change advisory boards that grant approval for deployment, are not stealthy and do not have the ability to be safely extended. As opposed to permanent host monitoring agents, MIRV follows the principle of temporary militarisation - additional forces are deployed to a compromised area and withdrawn after the breach is contained.

MIRV achieves this by offering the operator more introspection capabilities of the host. MIRV's main design feature are the embedded Lua micro-agents to monitor various system activity events and the ability to act on those events using the full flexibility and safety of Lua. Metasploit's meterpreter offers the volatility to ease the deployment concerns as well as stealth to observe attacker behaviour.

A mass-deployment plugin automates invocation of any existing metasploit module, such psexec by swapping the target host parameter with one from a list or range and automatically invoking it in background, allowing to take control of many machines quickly.

From there on, the operator may choose to deploy custom Lua scripts that either do a one-shot check or enter a loop and report to an event collection system in out-of-band fashion. MIRV exposes some crucial, but otherwise not so easy to obtain information sources:

- Windows logs presented as plain text.
- Terminal Services client share drive contents.

Additionally, easy to deploy and dynamically injectable application level hooks are made available.

This tool is aimed to be a practical proof of concept of a 'fightback' host agent. It currently works only on Microsoft Windows.

Introduction or preparing for the unexpected.

"Fortune favours the prepared mind" (c) L. Pasteur

A computer network security breach means that the preventive controls of the network have failed in one form or another. However, an incursion by intruders into the network does not mean that the intruders have successfully completed their mission. In a vast computer network like the networks of big, international companies, there may be many intermediary steps required before intruders even approach their information targets. This presents the defenders with an opportunity to repel the intrusion.

Defenders are faced with multiple challenges when fighting the intruders in this middle stage; in this paper, the middle stage includes the kill chain elements installation, C2 and actions on objectives as described in [1] and as opposed to early stages before an intrusion is made and post-mortem when actions on objectives have been completed and the remaining questions are "How" and "What's the damage?".

One of the most significant challenges is the agility asymmetry between the intruders and defenders. Intruders have very little restrictions in their actions or choice of tools while defenders have many, including, but not limited to: intruders changing tactics, organisational inertia, defender's dilemma and inadequate tooling which does not help to address these challenges.

This paper presents Metasploit Incident Response Vehicle (MIRV) - a proof-of-concept tool which is invoked from the Metasploit framework and aims to give defenders the agility to adapt to an intruder's evolving methods during the course of an intrusion campaign. In other words, MIRV helps prepare for the unexpected.

Defenders' challenges

The main advantage of an intruder is the agility asymmetry. Empirically, the main identified challenges are:

The defender’s dilemma:

As the defender’s resources are limited, how is it possible to monitor and act on the large number of systems? A mid-sized corporate network can easily reach 10,000 nodes and even if focus is on the key assets, they still may number in their hundreds. This is further complicated by a "fog of war" - "an intruder's idea of your network may not be your idea of your network". An intruder may have sometimes a better visibility of an organisation's network, for example, a

contractor's laptop which accesses information in the network is a valid target for the attacker while the defender may not even be aware of it. This is further exacerbated with Bring-Your-Own-Device (BYOD) policies in many workplaces. A success in overcoming the defender's dilemma would be to turn it into an intruder's dilemma as postulated in [2]: "The defender only needs to detect one of the indicators of the intruder's presence in order to initiate incident response within the enterprise." In practical terms, if the defender could detect a good Indicator of Compromise ("IOC") and then sweep the network, it may be possible to repel the intruder.

Organisational politics:

"Rome is burning! Yes, we understand that, but the earliest date a change-advisory board can convene is next Tuesday and we're in a change-freeze period now anyway..."
A computer network breach can have long lasting and detrimental effects on an organisation, ranging from loss of trust and resulting loss of productivity, loss of trade secrets or fines by regulators, etc. But in most cases, the exact loss is very difficult to project without very specific details of who are the threat agents, what is their motivation and what kind of access they have or might have? This heavily contributes to uncertainty which in turn often means lack of senior sponsorship, which in turn means that defenders will be held back by organisational policies designed for "peace time", such as slow change control. In some organisations, an emergency change can be applied not sooner than two business days. 48 hours without adequate intervention could be enough for intruders to accomplish their mission and depart.

OPS (Operations) mentality:

"The website is up, there is no problem."
The main criteria of successful operation of computer networks is often "is it working?" which means "are my files accessible, can I send and receive my e-mails and can I look at pictures of funny cats?" or in other words, the main emphasis is on the availability. Unless there is a major availability incident, the defending team is likely to be stalled by the operations maintenance team: in most cases, it will be necessary to introduce and run tools on key business systems. This means there is a chance of a disruption and it will be heavily resisted by the operations team as it negatively impacts their KPIs.

Stealth requirement

In addition, the defending team essentially runs a counter-intelligence operation, whereby they must collect enough information to shut down all access vectors the intruders may possess to prevent them from re-establishing themselves. If defenders reveal their position too soon, intruders may simply choose to go silent for a while.

Limitations of common tools

A computer network fortified against intrusions will have a robust security infrastructure deployed, which will include various network components, such as firewalls, centralised logging, SIEM systems, IDS, IPS, packet capture devices as well host components, such as HIDS, or advanced intrusion and anomaly detection agents.
More commonly, computer networks are not fortified and only have a firewall as a network level defence and an anti-virus solution to defend hosts. Essentially, a "temporary militarisation" is required. Defenders typically have only a few tools they can deploy quickly. Major limiting factors of quick deployment are availability of tools, complexity of installation and cost; cost can mean not only the absolute price in a selected currency, but also internal spending approval limits which may prevent quick purchase.
Tools that can be deployed quickly are commonly limited to an IDS and packet recording system (essentially whatever Security Onion has) at a network level, a log collection facility for select key assets and execution of scripts to gather information at the host level. In the author's opinion, the biggest gap in the quickly deployable tool arsenal is with host-based tools - clearly, having an advanced and flexible agent to monitor activities is useful, but deployment of most agent-based tools cannot be done quickly for the reasons stated above.

MIRV - a flexible and volatile agent

MIRV attempts to address the challenges and limitations described hitherto.
MIRV is a tool which is designed to meet and help overcome these challenges by introducing a massive, flexible (can be safely extended on the fly), volatile (operates in memory with minimal on-disk artefacts, gone after reboot) and stealthy (it is based on Metasploit's meterpreter).
These features hopefully will help overcome the challenges and limitations:

- The defender's dilemma is addressed by enabling a mass deployment of MIRV. MIRV does not attempt to report back every activity, but only "interesting" activity, where interesting is defined by the defenders on a case by case basis. Thus, even modest hardware could be used as an information collection and management node.
- As MIRV can be safely extended on the fly, there is a need to go through the gauntlet of change management approval more than once. If attackers change their methods, new patterns can be loaded into the tool.

The ambition is to take MIRV from a tool to an accepted concept and methodology.

It is true, that most Windows systems will have a similar agent already deployed - an anti-virus and through updates, perhaps, A/V could help defenders to adjust to specific intruders' tactics. However, most organisations do not have the ability to produce and push A/V updates based on threats identified internally. Also, A/V is not a stealthy system - attackers will almost immediately

be alerted if an approach that they had used for a while suddenly was blocked by A/V. With MIRV defenders can create custom signatures or rules, for example, if attackers use predictable filenames, such as ‘dumpMyActiveDirectory.exe’ or 'loot.rar', the creation of such files could be intercepted and defenders notified.

Technical overview

MIRV is in essence an extension of Metasploit and Metasploit’s meterpreter on the Windows platform. The basic principle is to deploy the extended meterpreter - MIRV on a significant proportion of systems using psexec or similar methods and then use the flexible monitoring and active response methods to detect attackers, collect information about them and then kick them off systems.
While nothing prevents deployment of new meterpreter extensions in the current version of meterpreter, these extensions are in form of DLLs written in C/C++. Lack of memory management and a safety framework around these extensions makes deployment of new extensions as a response to attacker’s activities a risky activity that may crash the deployed meterpreter or at worst - negatively impact the system it was deployed on. To address this issue, MIRV comes with built-in Lua scripting engine. Lua is an easily embeddable and easily extensible scripting language. In MIRV, multiple Lua scripts can be run as their own threads which allows for creating micro monitoring, alerting and processing agents. Some important activities, like windows log processing which requires complicated preprocessing to turn the logged data into familiar text representation, are too complicated to be handled entirely in Lua; therefore Lua code acts only as a filter and trigger and the heavy lifting is done in C. Also, system hooks must be written in C but can be extended in Lua.
Having such a flexible approach allows defenders to dynamically adjust their tooling in response to attacker activities.

Features

Features are roughly divided into two major sections - things driven by Lua and things filtered by Lua. Some emphasis is put on getting data that might not be available through conventional means, for example, if only successful or failed logins are recorded, then this information can perhaps be gleaned through other means.

Mass deployment

Metasploit console automation is used to deploy the MIRV/meterpreter agents. Given a list of computers and a set of valid credential, it uses psexec module to deploy agents. This creates only one artefact - the meterpreter executable which is deleted immediately after execution.

Lua micro agents

The core feature of MIRV is the Lua micro-agents. MIRV allows the operator to run multiple Lua scripts, each in its own thread. Primary aim of micro agents is to allow operators to do ad-hoc monitoring of systems. They follow a poll-filter-report cycle, for example, the script periodically polls process list and check if new and exciting processes have appeared while ignoring boring and mundane ones. An alert is sent with the new process names. To make these scripts richer, some Win32 Lua extensions are built-in.

Windows log processing

Getting windows logs to a central collection machine is not always a trivial task - the sheer amount of logs may overwhelm available resources, installing forwarders and making changes is a no-no. The logs also are not available in easy to read format. MIRV provides an easy way to get just the right amount of logs wherever necessary: It reads log sources and presents each entry to a Lua filter. If the filter lets it through, the log entry is sent to the remote destination.

RDP Session Hijacking

As an experiment into “hacking-back”, MIRV features the ability to execute processes in any RDP session’s context. If an attacker connects with his disks shared over RDP (the \\tsclient network path), the defender can use this to his advantage and obtain information about the attacker’s system or even attempt to compromise the attacker’s system by copying a backdoor into automatically executed locations.

[1] "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains". Eric M. Hutchins, Michael J. Cloppert, Rohan M. Amin, Ph.D. Lockheed Martin Corporation.
[2] "Defender's Dilemma vs Intruder's Dilemma", TaoSecurity, http://taosecurity.blogspot.co.uk/2009/05/defenders-dilemma-and-intruders-dilemma.html