Environment Aware: Future Directions

Version 1.0

Todd Heberlein
Net Squared, Inc.

27 Jan 2005

1 Introduction

This document describes the future directions for the ARDA sponsored Environment Aware Security project. We have been conducting research and development on this project for approximately 14 months, and this document reflects our adjustments to the original plan based on the results produced and knowledge gained during this time. The vision provides a framework for technology and capability that can grow for many years, but our focus is to create deliverable beta products that provide immediate value to the end customer by Summer 2005.

As originally envisioned, the Environment Aware Security project would consume detailed information about a network (e.g., vulnerabilities and topology) and a (potentially hypothetical) adversary's capability and produce (1) a set of systems that can be penetrated by the adversary and (2) a prioritized list of changes to the network (e.g., patches to specific systems) to maximally disrupt the adversary's ability to move through the network. The prioritized list of changes, referred to as Network Tasking Orders (NTOs), was envisioned as the primary way network and system administrators would interact with the system. Every day a network or system administrator would download a list of NTOs and, time permitting, work their way through as many items on the list as possible.

The plans in this document describe two separate but related efforts. The first effort, for the time being, is to reposition our original attack graph tool as a demonstration tool that shows the value of collecting detailed information about a network's configuration and usage patterns. Our concern is that the customer does not have the relevant data easily available that is necessary to perform a full attack graph analysis. At a later date the demonstration tool can be repositioned to an operational tool by a third-party contractor who can work closely with the customers to harvest the relevant data.

The second effort is to build a deployable Intrusion Containment System (ICS), with the first beta system available to the customer within six months. The concept for the ICS came out of our concern that the customers' networks do not readily deploy access control mechanisms within their networks, and if this is the case, there can be no optimal strategy for the customers to harden their networks. The heart of the ICS is the Intrusion Containment Appliance, a low-cost device that wraps a server and prevents it, if penetrated, from being used as a launch pad to spread the attack.

This document is divided into several sections. Section 2 describes our findings to date that have informed our future plans. Section 3 describes the two primary paths we plan to take our research and development. Section 4 describes the planned feature sets for the ICS. And Section 5 summarizes this document.

2 Findings

This section describes the relevant findings of our research for approximately the last 14 months. These findings include results from experiments, feedback on our ideas, and observations based on presentations and discussions with people at the recent ARDA PI meeting.

2.1 Research

Because we have not had access to the customers' networks or details about them, and probably would not throughout the remaining time of the project, we developed a tool to statistically grow a network based on an initial set of parameters. We used the tool to test attack graph analysis, and learned a number of facts that challenge our original plan. Our original plan would make heavy use of information from vulnerability scanners. In a sense, we would add value to the results provided by these scanners. However, we discovered the information produced from the standard set of vulnerability scanners is woefully inadequate to understand the risk posed by an adversary.

The following is a short list of many features that are critical to understanding the real risk posed by an adversary but are not available from standard vulnerability scanners:

- **Login information:** once an adversary penetrates a system, he can trojan login clients (ssh clients, web browsers, etc.) to collect passwords from users as they login to other systems. This is allows an adversary to penetrate additional systems even when those systems do not have vulnerabilities.
- **Client vulnerabilities and client/server connections:** Once a server is penetrated, if the adversary knows how to exploit a client's vulnerability he can attack the clients that regularly connect to that server. Full analysis of clients' vulnerabilities and which servers they connect to is critical to understanding the risks an organization faces.
- **Internal Access Control Lists (ACLs):** Without internal ACLs, an adversary in any part of an organization's network can penetrate any vulnerable service and much of the expected scale-free properties of an adversary's attack graph is lost. Deploying internal ACLs are critical to protecting a network (especially in the face of zero-day vulnerabilities), and knowing about them is critical to understanding an organization's risk exposure.
- **Wrapping servers:** One of the most effective ways to deploy ACLs internally and reducing the risks posed by known vulnerabilities and unknown vulnerabilities is to wrap servers to prevent them from making arbitrary outbound connection requests. Wrapping servers also change the exponential spread rate of a worm to a much slower linear spread rate.

To provide the value of wrapping servers, we developed and deployed within our own network a set of ACLs for Cisco routers (and routers and firewalls that support Cisco IOS rule sets).

2.2 Feedback

When initially briefing people about the Cisco ACL rules for wrapping servers, there were two primary criticisms. First, because of costs, organizations do not deploy network devices that are capable of supporting ACLs near servers (they often have simple switches and hubs close to end systems, including clients and servers). Second, organizations generally prefer to use routers only for routing and use firewalls for restricting traffic, and few organizations deploy firewalls beyond their perimeters. To address these concerns we identified low-cost network appliances that can be repurposed to support wrapping of servers. With the addition of low cost appliances for wrapping, everyone I have spoken to, including numerous people at the recent ARDA PI meeting, likes the idea and raised no objections.

2.3 Other Observations

In addition to the results from our own research and feedback at the PI meeting and other locations, several other observations are shaping our plans.

One of the first speakers at the recent ARDA PI meeting was Sherrill Nicely, Information Assurance Director, Community Management Staff. She described their networks as soft and chewy, an all too common description of networks in which the bulk of security prevention is directed towards the outsider. Once an adversary is inside the network (through penetration, as a consultant or guest, an employee, as a Trojan horse, etc.) movement inside the organization is fairly easy.

I have heard this "soft and chewy" description within the intelligence community before. For example, many years ago we were talking to the CIA about intrusion detection for databases, and I asked about the protection mechanisms provided by database vendors (e.g., permissions on tables or views). The response from the CIA employees was that protection mechanisms were often not used. A database would often be created for a specific project, and operational capability (get it running and get the results you want) was the primary priority. Security was considered less important in part because everyone had already gone through a vetting process and were therefore trusted users. This is the "everything runs at system high" concept.

At the ARDA PI meeting discussions of analyst and the sensors they used focused almost exclusively on analysts external to any organization (e.g., third-party Security Operations Center monitoring many sites) with virtually no discussion on analysts and analysis of intra-organizational activity.

Other discussions with IC analysts and employees indicated there was very little or no use of single sign on technologies. One of the primary benefits of single sign in technologies is they provide an audit/accounting trail of which users log into which systems from which clients. As discussed in Section 2.1, this data is critical to understanding the potential risks facing an organization from an adversary.

Comments made by some analysts at the meeting pointed to their interests in a cheap solution. This included upfront costs, operational and maintenance costs, and costs of additional labor required of their analysts. For example, after one presentation discussing the collection of new data and generation of more information, one person commented something to the effect, "Great. Just what we need: more stuff to look at."

These and other observations have led to several tentative conclusions (given our limited access to knowledge about the customers and their networks), including:

- The customers may not have extensive and readily available visibility and control of many of the assets (including usage patterns) within their organizations. If this is true, activating existing sensors, deploying additional sensors, and aggregating and managing the data into a single (potentially virtual) repository will require extensive infrastructure development.

- The customers represent knowledge organizations in which the primary activities of their knowledge workers involve taking in data and lower-level information and producing additional information. Furthermore, the specific activities of the knowledge workers are dynamic, often resulting in ad hoc internal projects supported by the information infrastructure (e.g., quickly standing up a database to support a particular activity).
- In order to gain acceptance, solutions should be lightweight in terms of initial cost, on-going costs, and labor needs, and a clear case must be made for the potential benefits.

3 Future Directions

Based on results from our R&D, feedback, and other observations, we believe that it is in the best interest of the customer to pursue two major forks of development. The first fork is an extension of our attack graph work, and we describe this in Section 3.1. The second fork is the development of the Intrusion Containment System (ICS) – a system we can deliver to the customer in the next six months and based on the low-cost server wrappers we have started experimenting with. Section 3.2 discusses the ICS.

3.1 Attack Graph

In this section we examine the first of two proposed “end products” for the customer: the attack graph tool. First we examine the rational as to why we chose this path. Second we look at the value it provides the customer. And third, we briefly describe the end product.

3.1.1 Rational

Our attack graph tool suite is designed to take as input a detailed description of a network (including vulnerabilities in client and servers, internal access control configurations, login patterns, and general client-server usage patterns), an adversary’s capabilities (e.g., skill sets), and initial conditions (e.g., set of systems the adversary initially controls) and generates a graph describing how the adversary can move through the organization’s network. The simple graph output can be of value in itself – it answers how deep into your organization an adversary can go. The graph (or a set of graphs generated by seeding multiple potential adversaries in the network) can be further processed to identify a prioritized list of changes to the network (the Network Tasking Orders (NTOs)) to maximally disrupt the adversary’s ability to move through the network.

Our fundamental concern is that the customers do not have easily accessible and adequately detailed knowledge of configuration and usage patterns within their networks necessary to build an accurate attack graph.

Because of this concern, we propose to reposition the current attack graph tool to a modeling and education tool that, at a later date, can be repurposed to an operational tool when the appropriate information is available. For example, a third-party contractor with appropriate security clearances could work with the customers to develop the necessary tools to harvest the information, thus moving the tool to an operational role.

3.1.2 Value Propositions

This approach provides three value propositions. First, it removes the uncertainty about how much access/support we will have to the customers' networks. We will assume we have none. However, in the future, contractors with the appropriate clearances can extend the approach to operational networks.

Second, the solution we will develop provides a quantitative and visual argument to the question: Why should the customer collect the additional information about their network? Collecting that information will not be free, and our solution will show that given an additional piece of information (e.g., login paths) how much more accurate an organization's predictions can be regarding their risks posed by an adversary.

Third, the attack graph tool will quantitatively and visually demonstrate the value provided by adding the Intrusion Containment System to their network. In other words, this tool will be used to sell the other tool, a tool that can be operationally deployed into customers' networks.

3.1.3 Proposed Attack Graph System

The primary focuses of the Attack Graph tool are to (1) provide quantitative and visual feedback as to the systems at risk from a given adversary and (2) demonstrate the level of precision with respect to correctly identifying the at risk systems based on the types of data available about the customers network.

The first focus primarily consists of putting a graphical user interface front-end on the individual command-line oriented tools and text files. From an experimental point of view, which has been our recent focus, command-line tools are preferable because they can be scripted to autonomously run large numbers of experiments and extract quantitative results. However, a graphical user interface provides additional value for demonstrations and individual, interactive experiments. The second focus primarily consists of restricting what information about the network (e.g., login patterns) the attack graph engine can use when identifying the risks posed by an adversary, storing the results from multiple runs using different levels of information about the network, and comparing the differences between the runs. Technically this simply requires adding switches that identify what information to test for in each stage of an attack to the attack graph engine.

Figure 1 shows the straw-man diagram for the system. (1) The user sets the parameters that describe a type of network in which they are interested. (2) The network description parameters are written to a file, and (3) the network growth engine reads in the configuration and probabilistically grows a network matching the description. (4) The description of the network is written to an external file. (6) The user describes the adversary of interest, the GUI writes out the description to an intermediate file that is read by the attack graph engine. (5) The attack graph engine takes the description of the network and adversary and (7) identifies the set of systems that can be penetrated by the adversary. This information is then displayed on the user interface. Step (8) will allow the user to tweak the network description (e.g., adding or removing firewall rules).

Step (9) shows how a third-party contractor can turn the system from a demonstration and experimentation tool into an operational tool by collecting the relevant information about the customers' networks and saving that to the network description repository.

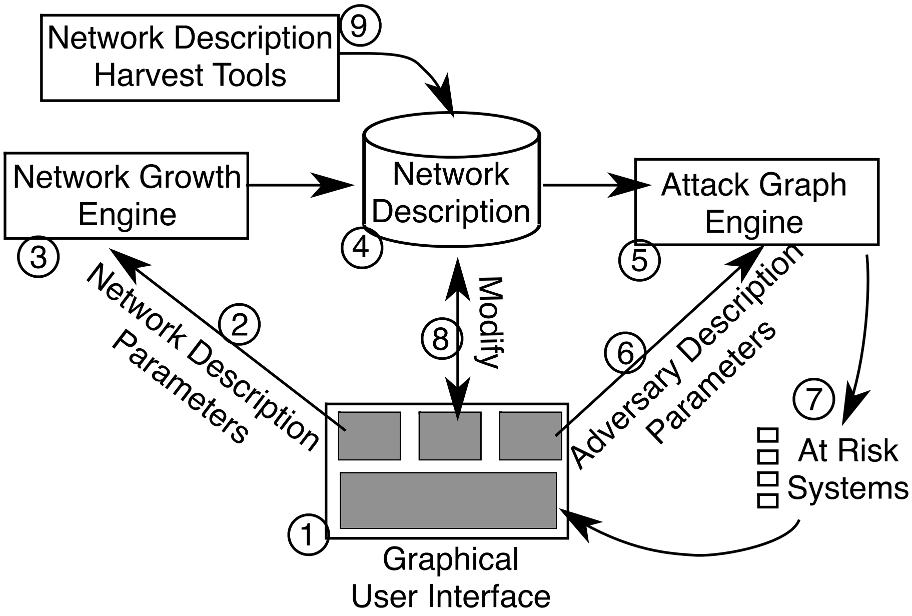

Figure 1: Attack Graph Tool's Data/Control Flow

Of this proposed system, steps (2), (3), (5), (6), and (7) are already operational. The primary work is to provide a GUI to allow the user to interactively configure the network and adversary descriptions and to modify the network description. Also, we need to formalize step (4), the external representation of the network.

3.2 Intrusion Containment System

In this section we examine the second of two proposed "end products" for the customer: the Intrusion Containment System. First we examine the rational as to why we chose this path. Second we look at the value it provides the customer. And third, we briefly describe the end product.

3.2.1 Rational

One of our original assumptions for our Environment Aware project has been that the graph representing how the adversary can move through a customer network exhibits scale-free behavior. A scale-free network's nodes connectivity rate have a power-law distribution. That is, a small number of nodes are highly connected while most nodes exhibit very little connectivity.

Because of the power-law distribution, a scale free network is robust against random attacks (since most nodes attacked have very limited connectivity) but highly susceptible to targeted attacks against the small number of highly connected nodes. For our purposes, this means that we can maximally disrupt the adversary's attack graph (i.e., his ability to move through a customer's network) by focusing on the super-nodes in the adversary's attack graph. This is the fundamental rational behind the concept of the

Network Tasking Orders (NTOs) – the prioritized fixes to the network focus first on the super-nodes in the adversary's attack graph.

Our fundamental concern is that attack graphs for the customers' networks are not scale-free, so there exists no prioritized fixing scheme to maximally disrupt the adversary's ability to move through the network.

The reason the adversary's attack graph may not be scale free is that, if casual discussions with customers about their networks are accurate, there is little use of access control rules inside the network (that is, they are "soft and chewy"). Without internal access controls, every node inside the network has the potential to connect to every other node, and instead of a scale-free network the adversary's attack graph is a fully connected network. Not only does this mean that there is no prioritized scheme for hardening the network, but the network is optimally designed for the adversary.

To address this concern we plan to develop the Intrusion Containment System (ICS), a system inspired by our experiments over the last six months. The ICS provides a lightweight means to deploy internal access controls inside the network by wrapping servers with a low-cost appliance (the first version will be based on a Linksys router discussed at the ARDA PI meeting). Not only does the ICS provide immediate value on its own, but also it moves a potential adversary's attack graph from a fully connected network back to a scale-free network. This means that a prioritized hardening scheme, the Network Tasking Orders, become a viable solution.

3.2.2 Value Propositions

The ICS provides a number of values to the customer. First, the ICS addresses the problem of a fully connected adversary attack graph. As previously mentioned, if the customer does not deploy internal access control limitations inside their network, an adversary's attack graph is not scale-free but fully connected. By deploying appliances to wrap servers the ICS transforms adversaries attack graphs from fully connected to scale-free. This not only reduces the number of systems that can be successfully attacked, but it also allows the attack graph analysis system to identify an optimal strategy to harden the network.

Second, the ICS helps address the problem of unknown vulnerabilities. Assuming the adversary is sponsored or supported by a state or well-financed organization (funded terrorist group, organized crime, etc.), we should expect the adversary to have zero-day attacks – attacks against unknown vulnerabilities for which no patch exists. Our experiments over the last six-month period demonstrate that the ICS approach greatly limits the damage that can be done by such attacks.

Third, the ICS slows a server-based worm's spread rate from exponential to linear. Experiences with server-based worms such as Code Red and Slammer have demonstrated how quickly these worms can spread, and theoretic worms such as Warhol or Flash worms can greatly accelerate the process. Any active worm defense must respond incredibly quickly in order to stop or slow such worms. The worms achieve their exponential spread rate by co-opting each victim to become an attacker as well. By deploying the server wrapper appliances we prevent the servers from becoming attackers as well, so the spread rate is dramatically slowed down. This in turn provides additional time to deploy a successful response strategy.

Fourth, the ICS uses low cost appliances, so entry costs to the organization is kept low. The first version of the server appliance wrapper is based on a modified low cost ($50-$70) Linksys router.

Fifth, the ICS has a very low operational cost. Signature-based intrusion detection systems and vulnerability scanners must be continually updated as new vulnerabilities and attacks are discovered. The ICS appliance is a "deploy and forget" technology. Because the legitimate needs of servers to make outbound connections rarely changes, the server wrapping appliances have a very low maintenance cost.

Sixth, the ICS also provides an intrusion detection capability. If an adversary penetrates a server and then attempts to make an outbound connection, that action will be detected and reported to the ICS management station. Subsequent server wrapper appliances could potentially reroute the offending outbound connection attempt to a honeypot so the attack can be captured and observed in a controlled environment.

3.2.3 Proposed Intrusion Containment System

The Intrusion Containment System consists of four types of devices; although, only two of them are necessary and will be part of the first release. Figure 2 shows the strawman diagram for the ICS. (1) The Intrusion Containment (IC) Appliance wraps the server. (2) The IC Management Station manages the IC Appliance, and the IC Appliance sends reports to the IC Management Station. (3) The IC Management Station is where a network administrator manages the IC Appliances throughout the network and accepts reports from the optional IC Sensors and Honeypot. (4) Normally the IC Appliance would block an unauthorized outbound connection request, but optionally the IC Appliance could reroute the outbound connection to a Honeypot in order to capture a potential attack vector. (5) When the Honeypot captures an unauthorized outbound connection, it is quite possibly an attack; the Honeypot then forwards the presumptive attack to the IC Management Station. (6) The ICS can also take advantage of additional sensors that could detect active servers that are not wrapped by an IC Appliance, or capture unusual inbound traffic to servers that can be saved if the IC Appliance later detects an outbound connection attempt, Each element is described in the next several sections.

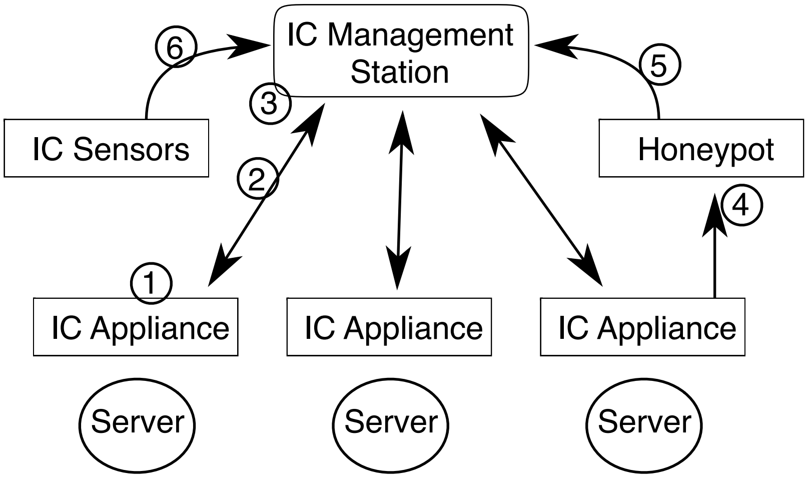

Figure 2: Intrusion Containment System Architecture

3.2.3.1 IC Appliance

The IC Appliance is the heart of the ICS. It wraps the server, allowing inbound connections but restricting outbound connection to a well-defined set of destinations. The owner of the server manages restrictions of inbound connections – in other words, access control to the server is discretionary. The network administrator manages restrictions of outbound connections – that is, outbound access control is mandatory. It is this balance of discretionary access delegated to the owner of the server and mandatory access managed by the network administrators that allows the knowledge worker to carry out his mission without endangering the security of the rest of the network.

While most operating systems today include firewalls, these can only be trusted to protect against inbound connections. If the server attempts to use outbound firewall rules, the adversary, once he controls the server, can simply change the firewall rules to allow the outbound connections. This is why Mandatory Access Control rules must be managed by a device outside the control of the server's operating system. Hence the need for the appliance.

The IC Appliance will operate in one of five modes:

- **Pass-through:** The IC Appliance operates transparently allowing all packets in both directions to pass freely.
- **Learn:** The IC Appliance allows packets to freely flow in both direction but learns which outbound connections the server typically makes.
- **Pass-through and Report:** The IC Appliance allows packets to flow freely in both directions, but outbound connections that violate the access control rules are reported to the IC Management Station.
- **Block and Report:** The IC Appliance blocks unauthorized outbound connection attempts and reports the activity to the IC Management Station.
- **Redirect and Report:** The IC Appliance redirects outbound connection attempts to the Honepot and reports the event the IC Management Station. This feature is not currently supported in our Cisco IOS implementation.

Our first implementation of the IC Appliance was developed using a Cisco router using the IOS Access Control List rules. Unfortunately a Cisco router is relatively expensive. We determined a low-cost Linux solution would suffice in most instances, and we are currently modifying a low-cost Linksys router (which is based on Linux) for the first version of the IC Appliance.

3.2.3.2 IC Management Station

The IC Management Station provides the graphical user interface that the administrator uses to control and monitor all the IC Appliances and any additional IC Sensors or Honeypots. This will simply be software that can reside on the network administrator's workstation.

3.2.3.3 IC Honeypot

The IC Honeypot is an optional feature to capture any unauthorized outbound connection attempts. An unauthorized outbound connection may indicate that the server has been compromised and the adversary is attempting to penetrate additional systems or exfiltrate information. By redirecting the attempted connection to a honeypot the administrators can capture the possible attack attempt (which may be new and previously undetectable) or the information the adversary attempted to exfiltrate.

In general, a readily available honeypot could probably suffice; however, for safety's sake, the honeypot should probably be wrapped with an IC Appliance as well.

3.2.3.4 IC Sensors

IC Sensors provide additional optional features for the ICS. The most important sensor is one that would monitor for active servers and compare them to the list of servers wrapped by IC Appliances. The IC Sensor would report to the IC Management Station any detected server that is not wrapped by an IC Appliance.

A second optional IC Sensor could be a network flight recorder – a device to record network traffic for short periods of time. If a given amount of time elapses without any indication that the data is important, the data is deleted. An indication that the data is potentially important could be a report from an IC Appliance that a server attempted to make an outbound connection. Such a report could indicate that the server was recently compromised, and the data from the network flight recorder may have recorded the initial and previously undetected penetration.

Additional servers are possible, but care must be taken not to generate too much data for the network administrator to process.

4 Deliverables For The ICS

The Intrusion Containment System (ICS) is the primary near-term deliverable we expect the customer to deploy within their organization. This section identifies the planned feature sets for ICS Version 1.0 and Versions 2.0 and beyond. A beta for Version 1.0 is planned for the Summer 2005 ARDA PI meeting.

4.1 Version 1.0

The section lists the features planned for the various ICS components for the first release of the ICS. A beta with these features will be available for the Summer 2005 ARDA PI meeting, and a full 1.0 release will be planned at the end of the ARDA contract.

4.1.1 IC Appliance

- Remote configuration: Add and remove filtering rules and set operational mode from the remote IC Management Station.
- Collect and supply statistics: Collect basic statistics such as uptime and packet flow, and provide these statistics to the IC Management Station in response to a GET command. This can also be used as a basic heartbeat mechanism.
- Two-way authentication: The IC Management Station authenticates itself to the IC Appliance, and the IC Appliance authenticates itself to the IC Management Station.
- Pass-through mode: Allow all packets to flow freely in both directions.

- Pass-through and report mode: Allow packets to flow freely, but report to the IC Management Station connection attempts that violate the filtering rules.
- Block and report mode: Block connection attempts that violate the filtering rules, and report to the IC Management Station the blocked activity.

4.1.2 IC Management Station

- Remote control and monitoring of IC Appliances.
- Two-level management: Support and coordinate the management of IC Appliances by both a local department and a network operations center (NOC). Thus, a local department system administrator can deploy an IC Appliance for a local server, and the organization's NOC will be appropriately notified of the new appliance.
- Database of known servers: Track known active servers within the organization, including identifying which servers are currently wrapped by IC Appliances.
- Two-way authentication: The IC Management Station authenticates itself to the IC Appliance, and the IC Appliance authenticates itself to the IC Management Station.
- History alert database: Track reports from IC Appliances.
- Common alert explanation database: Keep a database of common alerts and the potential causes, both legitimate and illegitimate, for the alerts. Many servers will attempt to make legitimate outbound connections for activities such as DNS, NTP, user authentication, and software updates. There may also be a number of well-known common outbound connections made my adversaries (e.g., a worm trying to download additional code). When an IC Appliance sends a report to the IC Management Station, before displaying the report to the analyst, the IC Management Station consults this database in order to augment the alert report with a possible explanation for the cause of the alert.

4.1.3 IC Sensors

No IC Sensors are planned for Version 1.0 of the ICS.

4.1.4 IC Honeypot

No IC Honeypot is planned for Version 1.0 of the ICS.

4.2 Version 2.0+

This section lists additional features planned for the ICS components for version 2.0 and beyond. Some of the features will be available in a 2.0 beta demonstrated at the end of the ARDA project.

4.2.1 IC Appliance

- Pass-through and learn mode: Allow all packets to flow through, but learn outbound connection attempts.

- Redirect and report mode: Redirect unauthorized outbound connection attempts to a designated honeypot, and report the event to the IC Management Station.
- Simple inbound intrusion prevention: Drop inbound packets containing data that matches a small number (1-5) of simple signatures. The purpose of this capability is to protect the wrapped servers from active attacks for which a patch is not yet available.

4.2.2 IC Management Station

- Vulnerability database: Integrate a vulnerability database (e.g., provided by Nessus). When an IC Appliance reports an unauthorized outbound connection attempt, the IC Management Station can, automatically or on demand, augment the report with information about the offending server from the vulnerability database.
- Intrusion report database: Integrate an intrusion detection alert database (e.g., those provided for Snort alerts). When an IC Appliance reports an unauthorized outbound connection attempt, the IC Management Station can, automatically or on demand, augment the report with information about any recent intrusion detection alerts targeting the offending server.
- Passive server detection: Integrate reports from an IC Sensor that detects active servers through passive monitoring. The purpose of this feature is to identify unknown servers that are not wrapped by an IC Appliance.
- Active server detection: Integrate reports from an IC Sensor that detects servers through active monitoring. The purpose of this feature is to identify unknown servers that are not wrapped by an IC Appliance.
- Network flight recorder. Integrate reports from an IC network flight recorder.
- Honeypot: Integrate reports from an IC Honeypot.
- Network forensics tools. Add tools to perform forensics analysis on the network data captured by IC Sensors such as network flight recorders and honeypots.

4.2.3 IC Sensors

- Vulnerability scanner: Use or modify as necessary existing vulnerability scanners such as Nessus; send reports to the IC Management Station.
- Intrusion detection sensor: Use or modify as necessary existing intrusion detection sensors such as Snort; send reports to the IC Management Station.
- Passive server detection sensor: Look for servers by observing their traffic (e.g., a server sending a SYN-ACK back to a client). Send reports of newly detected servers to the IC Management Station.
- Active server detection sensor. Look for servers by actively probing hosts in the network (e.g., sending connection requests to various server ports on the machines in the network). Send reports of newly detected servers to the IC Management Station.
- Network flight recorder: Collect and store packets for short periods of time, and only archive the data should an appropriate event occur (primarily an IC

Appliance reporting an unauthorized outbound connection attempt). This data can then be analyzed by forensic analysis tools to potentially identify the underlying cause of the unauthorized connection attempt.

4.2.4 IC Honeypot

- Honeypot sensor: Use or modify as necessary an existing honeypot to capture unauthorized outbound connection attempts that have been redirected by an IC Appliance. Send a report (and possibly data) of the captured session to the IC Management Station.
- Proxy server sensor: The honeypot sensor not only accepts a redirected connection request of an unauthorized outbound connection, but it also forwards the connection to the original destination. Thus the honeypot behaves as a man-in-the-middle proxy server, allowing the connection to continue but capturing and controlling the data flow. This can allow (1) legitimate activity to continue to operate normally from the offending server's point of view, (2) analysts to capture more information about the attack (e.g., capturing an additional phase of an attack), and (3) using content analysis technology (e.g., string matchers) to block sensitive data from being exfiltrated.

5 Summary

This document describes our plans for future research and development on the Environment Aware Security project. We have already developed several attack graph engines (SMV, JESS, CLIPS, and a homegrown C++ version), and we have developed a tool to grow simulated networks based on a range of parameters used to describe a potential customer's network. We have run a large number of simulated attacks with various adversary models on different network types.

We also developed and deployed a set of access control rules for a Cisco router (or any router, switch, or firewall that supports the Cisco IOS API) that can wrap a server and prevent it from making unauthorized outbound connections. We have started exploring the use of a low-cost appliance to provide a similar capability. When deploying such a capability in a network, our simulations show a dramatic reduction in the number of systems that an adversary can penetrate.

We have also taken feedback of our ideas at the recent ARDA PI meeting, and we heard from a number of people describing features of the customers' networks.

These three lines (simulated experimental runs, server wrapper prototype, and feedback and discussions about the customers' networks) have led to the plans discussed in this document. The plans outlined in this report present a two-pronged development strategy for the Environment Aware project.

The first effort is to continue development of our attack graph tool suite, but instead of aiming for an initial operational deployment the effort is focused on demonstrating the quality of analysis that can be performed based on the level of information available to the analysis engine. The goal, in part, is to convince the customer of the value of collecting the relevant information (e.g., login patterns). A third-party contractor could work with the customer to collect the relevant information and transform the attack graph tool from a demonstration system to an operational system.

The second effort is to build the Intrusion Containment System (ICS). The heart of the ICS is the Intrusion Containment (IC) Appliance, a low-cost device that wraps servers and prevents them from making outbound connections. An IC Management Station will be used to manage the IC Appliances throughout the network. In addition to the IC Appliance and IC Management Station, future versions of the ICS could support optional IC Sensors and IC Honepots to capture additional details about new attacks or exfiltration attempts. The ICS is designed to be inexpensive to deploy and easy and inexpensive to operate.

There is also a synergy between the two efforts – the attack graph demonstration tool can show the value of the ICS by quantitatively and visually demonstrating the reduction in risk to a customer’s network after deploying the ICS.