Network Traffic Proxy System

Interview Report

Version <1.7>

11/04/2018

# Document Control

## Approval

The Guidance Team and the customers, Dr. Jaime Acosta, Christian Murga, and Caesar Zapata will approve this document.

## Document Change Control

|  |  |
| --- | --- |
| Initial Release: | Version 1.0 |
| Current Release: | Version 1.7 |
| Indicator of Last Page in Document: | ∎ |
| Date of Last Review: | 11/05/18 |
| Date of Next Review: | 11/05/18 |
| Target Date for Next Update: | 11/05/18 |

## Distribution List

This following list of people will receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Elsa Tai Ramirez

Jose Cabrera Maynez

Customer: Dr. Jaime Acosta

Christian Murga

Caesar Zapata

Software Team Members:

Oscar Galindo

Isai Gonzalez

Kevin Gonzalez

Julio De La Cruz

Alan Caldelas

## Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 9/23/18 | Julio De La Cruz | Added questions and filled in details for sections. |
| 1.1 | 9/25/18 | Kevin Gonzalez | Finish the remaining sections and review the overall report in order to submit this Thursday on 09/26/18. |
| 1.2 | 9/26/18 | Oscar Galindo | Added transcription from minutes 25 to 40. |
| 1.3 | 9/26/18 | Julio De La Cruz | Finished transcription, filled out actions’ items section. Add to additional information section.  Revised question section. |
| 1.4 | 9/26/18 | Alan Caldelas | Adding additional references and additional information on the next steps |
| 1.5 | 9/26/18 | Isai Gonzalez | Finished purpose section |
| 1.6 | 11/04/18 | Kevin Gonzalez | Went over and addressed sections 2 and 3 comments left by the reviewers. |
| 1.7 | 11/04-05/18 | Oscar Galindo | Revised the document and table of contents. |

Table of Contents

[Document Control II](#_Toc529128966)

[Approval II](#_Toc529128967)

[Document Change Control II](#_Toc529128968)

[Distribution List II](#_Toc529128969)

[Change Summary II](#_Toc529128970)

[1. Purpose 5](#_Toc529128971)

[1.1. Project Overview 5](#_Toc529128972)

[1.2. Background and Contact Information of Interviewees 5](#_Toc529128973)

[1.3. Interview Information 5](#_Toc529128974)

[1.4. References 5](#_Toc529128975)

[2. Interview: Questions and Responses 6](#_Toc529128976)

[2.1. Prepared Questions and Responses 6](#_Toc529128977)

[2.2. Additional Information 9](#_Toc529128978)

[3. Action Items 11](#_Toc529128979)

[3.1. Inconsistencies 11](#_Toc529128980)

[3.2. Unclear Items 11](#_Toc529128981)

[3.3. New Questions 11](#_Toc529128982)

[3.4. Unanswered Questions 11](#_Toc529128983)

[3.5. Next Steps 11](#_Toc529128984)

[4. Appendix A 12](#_Toc529128985)

[4.1. Presentation Transcript: 12](#_Toc529128986)

[4.2. Q&A Session Transcript 14](#_Toc529128987)

[4.2.1. General Questions Transcript 14](#_Toc529128988)

[4.2.2. Packet Questions Transcript 14](#_Toc529128989)

[4.2.3. Hooks Questions Transcript 16](#_Toc529128990)

[4.2.4. Protocols Questions Transcript 17](#_Toc529128991)

[4.2.5. IPtables Questions Transcript 17](#_Toc529128992)

[4.2.6. Fuzzers Questions Transcript 17](#_Toc529128993)

[4.2.7. Users Questions Transcript 18](#_Toc529128994)

[4.2.8. Display/Interface Questions Transcript 18](#_Toc529128995)

[4.2.9. Security Questions Transcript 19](#_Toc529128996)

[4.2.10. Development Constraints Questions Transcript 19](#_Toc529128997)

[4.2.11. Extra Questions/Follow-UPs Transcript 19](#_Toc529128998)

# Purpose

This document will allow us to reflect on the interview and state what we got out of it. This document will contain the answers to the interview questions, any new questions or concerns that Team 5 may have regarding the project, and a transcript of the interview. The client can use this document to see what we understood and correct any misconceptions that we may have had about the project.

## Project Overview

The Network Traffic Proxy System is a system that will be able to intercept, modify, and forward packets, along with other functions. This system will be used to test security on a network and make that process as simple as possible. While there is already plenty of software that can do some of these specific actions, this system will be able to work in a more general sense and with many different kinds of packet layering. The system can use the software that was previously mentioned to assist in getting the job done. All of the functions should be in a convenient graphical interface that any intermediate to expert level analyst can use.

## Background and Contact Information of Interviewees

According to a memo sent by Dr. Elsa Tai [1], Dr. Jaime C. Acosta graduated with a Ph.D. from UTEP and currently works for the U.S. Army Research Laboratory and leads cybersecurity programs. Dr. Acosta also teaches, works with students, and hosts workshops at UTEP.

Christian Aaron Murga is currently pursuing his Master of Science at UTEP. He has worked with Google, the Smithsonian Latino Center, and the El Paso Intelligence Center, helping with the development of different applications. He has also worked with the Army Research Lab in creating cyber security curriculum workshops. [1]

Mr. Ceasar Zapata graduated from UTEP with a BS in Computer Science. He works for the U.S. Army Research Laboratory “as a Computer Scientist and Cybersecurity Analyst [1].” He is also involved in research projects and recruitment efforts. [1]

## Interview Information

Interview took place Thursday, September 20 2018, from 7:30 AM to 8:15 AM. The interview was subdivided in concern sections. For every section a team selected a representative that acted as the interviewer for a particular section. After every question the clients and the interviewers had the opportunity to make immediate follow-ups to their respective responses or questions. At the end of the Q&A session the group of interviewers and the public present had the chance to ask their individual questions to the client. Since the client was represented by two members these members complemented their answers as they saw necessary during the interview. All the interview was recorded, and the clients were welcoming of recording the interview. Elsa Tai was the conductor of the session, and she marked down the times for every representative to talk, she contributed to the discussion by adding questions for the client, and ultimately finalized the interview.

## References

[1] Elsa Tai Ramirez, private communication, Sep. 19, 2018.

[2] J. Acosta, C. Murga, C. Zapata, “Network Traffic Proxy System,” 2018.

# Interview: Questions and Responses

A list of all questions and answers that were provided during the interview with the client can be found in this section. Some of the answers will have additional information that was added to the questions after the conclusion of the interview and the review of additional sources. Following the questions and responses section will be a section dedicated to additional information, where a list of sources and other relevant information pertaining to the project can be found.

## Prepared Questions and Responses

**GENERAL**

* Please describe the different uses of the system.

A live method where we essentially become, or somehow get between the communicating nodes while traffic is live and intercept. Choose a specific packet modify it and move it along. Also, can use a PCAP file or saved capture and modify it as well.

* Please describe the current process used to test protocol vulnerabilities*.*

The current requires the manual development of specific scripts to perform all the desired modifications and forward of those packets we intercept and modify.

**PACKETS**

* What’s your motivation for wanting to use AFL for fuzzing when Scapy also supports it?

AFL is a smart fuzzer that can do more detailed modifications than Scapy. There are tradeoffs, but the analysts are more familiar with it. Even though Scapy will know a little more about the protocol itself, AFL is probably better.

* What can be modified in a packet?

Anything within the packet should be modifiable, if needed you should leverage solutions like tshark for being able to modify the dissected information. The process works by utilizing a dissector to dissect the intercepted packets at the kernel level. Everything that can be dissected should be modifiable.

* How should the system handle packet dissection?

Use dissectors. This will usually depend on the packet intercepting software not necessarily our system.

* How should the system handle encrypted packets?

Depends on the dissector being used, also use hooks by knowing encryption factors

* How should the system handle the priority level of multiple incoming packets?

Just as they arrive in the queue, priority is set in FIFO implementation.

* Would it be necessary to keep track of the changes done to the packets?

No, but we do want a save for the PCAP of all the packets

* In regard to Packet Data and the separation of it into layers (Ether, IP, TCP, ICMP, etc.,.), should the system be able to handle all variations of data packet formats?

Yes, it will depend on the dissector being used.

* Would packet analysis be done on the fly using hooks to decide how to modify the intercepted packets or would the modifications be decided by the user in real time?

The system should be able to use hooks to modify packets, but the analyst should also be able to modify the packet how they want in real time.

* What would an interactive modification of the packets entail? Does it mean that the application should be able to identify the protocol, the available modifications for each protocol, and then prompt the analyst for the desired modifications? Or should the software react to each different protocol by a set of pre-defined rules?

Either live went a packet is intercepted or through a PCAP file that stores previously captured packets.

* What options need to be present in the filter that selects which packets are going to be intercepted? What does it mean for a packet to be intercepted? Can there be multiple filters?

A display filter selection functionality should be present, the either allow you to intercept according to protocol, port, etc.

* What does it mean to modify, forward, and drop selected packets? When packets are dropped, where do they go? Are they deleted or archived in a history?

The analyst should be able to modify packets and forward them through the network. Packets that are dropped are completely removed.

* What conditions do the packets need to meet in order to get captured?

Packets should meet filtering conditions and hook logic.

* What conditions do intercepted packets need to meet in order to get send or dropped?

The analysts choose, but should have to ability to send or dropped packets

* When the packet interception mode is off, how should the system handle packets that are previously intercepted and in the queue?

Forward everything out of the queue

* Packets that are meant to be translated from IPv6 to IPv4, should the system intercept and translate the packets? If so, which tunneling method should the system support?

However the OS handles the translation.

**HOOKS**

* What programming languages are the hooks written in?

Python

* Should the system provide the analyst a way to create hooks on top of loading hooks? If yes, please provide a use case scenario on how hook would be implemented in the system.

No, our system should just provide a way to load hooks.

* How does the analyst define the conditions that cause the execution of the hooks?

The logic is defined through Display filter and logic in the hook.

* Please provide a use case scenario on how to create and store a hook collection. Several hooks that pertain to a specific protocol.

Hooks that serve a similar purpose, or look for a similar protocol, should be able to be bundled together into a hook collection. For example, certain hooks can be arranged by the kind of modifications they perform and what fields/logic they contain.

* Please provide examples of conditions and statues that can trigger hooks.

TCP.port = 80, TCP.port = 8080. It could also be any field in the packet, not just a port.

**PROTOCOLS**

* Burp Suite is a software system that currently performs a similar task, except that it only supports the Hypertext Transfer Protocol (HTTP). Should the system support any type of protocols including proprietary protocols?

Yes; the possibly dissectible protocols depend on the dissector capabilities.

**IPTABLES**

* Please provide a use case scenario on how to set up an iptables rules.

Look at the source code of Scapy, or similar software, to see iptables rules.

* What sort of filters would you like to specify to determine which packets are intercepted?

Display filters that should be described using BNF.

* Please provide a use case scenario on how to configure a filter to intercept live traffic.

At a very high-level description the analyst Starts system, specifies the filters, and then clicks intercept.

* The system requires analysts to set rules for iptables and save configurations, but should we anticipate the possible use of ip6tables(IPv6), ebtables(Ethernet Frames), arptables(ARP)?

These changes are possible in the future but should not be a main concern of system right now.

**FUZZERS**

* Should the project be able to fuzz packets on the fly or is fuzzing expected to be a process of its own?

It’s a process on its own but configuring the fuzzer should be able to be done live and within the system.

* Should the packets generated by the fuzzer count towards the queue of packets or is the fuzzing process separate from the other processes?

It should be a separate process.

* Since we are trying to fuzz packets and these packets might be completely unrelated to one another, in what way would the fuzzer test its mutations? If the mutations are not to be tested should the generated mutated packages be saved in a PCAP file for individual evaluation?

Yes, the fuzzed packets should be saved in a PCAP file for future revision by analysts.

**USERS**

* In the RDD, it stated that the system should be usable by intermediate to expert level analyst. Please define the difference between an intermediate level analyst and expert level analyst regarding their usage of the system.

Intermediate and above is someone that’s familiar with networking technologies, they have a understanding of how nodes communicate. What separates the advanced from the intermediate is that the advanced would know, how dissectors work, they’ve written a few of their own, they know about specifics about the protocol and what’s required for the connection to take place. TCP which is a standard, has to establish a handshake which is a three-way communication before communication can occur, so something like that or other protocols would be able to separate an advanced user from an intermediate, someone that can write hooks, because they have that knowledge would be more of an advanced user.

* Should the system generate logs containing user activity?

Would be nice, but not needed in this system

**DISPLAY/INTERFACE**

* What display modes are needed besides raw hexadecimal and decoded type?

Also include a Binary view that is able to handle with smaller than bytes fields.

* What does it mean for a type to be decoded? How the system should handle a type decoding failure?

This pertains to the dissector; the dissector knows how fields are arranged within the packet. If dissection fails, however, the system should notify the user, the system should not crash if dissection fails.

* How should the queue of intercepted packets be displayed in the GUI?

This is a design choice open for the developers of the system.

* Are there any specific GUI elements that exist in other tools we should consider when designing the system?

Yes, please take a look at other software like Wireshark, burp suite, and zapp.

**SECURITY**

* Do we need to take into account any special security features in the software’s design? Such as needing to work on both contractor and civilian computers, handling classified or sensitive information, or running in labs with an over the air barrier (no internet access of any kind, and no outside software allowed).

No, not our responsibility, it will be handled at a different level of abstraction within the terminal. The system should execute any modification instructions under the profile of a non-trusted user to avoid any possible exploits.

* Should the system support user authentication?

No.

* Should data generated by the system be encrypted?

No.

**DEVELOPMENT CONSTRAINTS**

* What are the constraints in regard to programming languages, such as the version and GUI libraries? Scapy supports both python2 and python3. Python2 will not be supported beyond January 1st, 2020. Is it acceptable to use python3 for the application?

Python3 is fine.

* In the RDD, it is stated that the system should support queues of at least 100 packets without any behavioral performance issues on a modern laptop system. Please define what constitutes a modern laptop and any performance requirements.

7th gen core i5 8 gb of ram.

* What operating systems should the system support?

Kali Linux/ kali 2018.2.

* Scapy has a GPL license. That means, in effect, anyone with an executable of the application needs to have access to its source code, including other DoD organizations and contracting firms. This also means that any derivatives of this software need to have a GPL license. Is that acceptable?

Yes.

* For hooks, it can be fulfilled using Amazon’s AWS Lambda resource or Microsoft’s Azure functions. Would you be open to using Lambda or Azure for this system?

No.

## Additional Information

In this section we are listing additional information that is relevant to our project. Some of the information included are additional quotes from the interviewees and links to websites that contain information about fuzzer, Scapy, NFQUEUE, Linux, Burp suite, Ping and other relevant websites. These notes are a summarized version of specific highlights that we consider relevant from in the interview with the client.

* “So, the very top layer is essentially what pertains to the application, in this case it’s the web server.”
* “...the upper layer uses the bottom layer as sort of a service and same way on the other end when its being deconstructed for reading the lower layer is used by the upper layer.”
* “The very top layer you usually have your application data, in this case its http but it can be something like an SSH server, skype data. The layer right below that which is called the transport. Is where you have things like, do I need this communication to be controlled, do I want to make sure every packet is received, and I get an acknowledgement for that packet. The layer below that is used for routing, so you saw that it had to travel the packets had to travel through several routers that's what that layer is used for. This is where you will see thing called IP address, Internet protocol addresses. The layer below that has to do with the hardware, the physical layer.”
* “...each of these layers we have things called protocols. So, this is where you can specify how the communication occurs for that particular layer.”
* “in the form of request for comments or RFC’s… These describe very explicitly how that protocol works and what has to go into that protocol for things to communicate properly.”
* “We want to use a Linux system… Kali Linux is a pretty good candidate...”
* “The main reason is for security testing. So essentially we need to be able to check for developers that might try to just hide things that are weak in their code.”
* See http://www.icir.org/vern/cs161-sp13/notes/Networking-terminology.txt for terminology definitions.
* Public protocol specifications are published and made available as Request for Comments (RFC) documents (e.g., see <https://www.ietf.org/rfc.html>)
* Scapy (see <https://scapy.net/>)
* NFQUEUE is a Linux technology that allows intercepting packets at the kernel level (before they reach operating systems services.) https://home.regit.org/netfilter-en/using-nfqueue-and-libnetfilter\_queue/ for an example of C and NFQUEUE used together and https://gist.github.com/eXenon/85a3eab09fefbb3bee5d for an example of Scapy and NFQUEUE used together)
* Ping is a program that works by using the ICMP protocol. Most operating systems have a ping program pre-installed and can be run from a terminal (see <https://en.wikipedia.org/wiki/Ping_(networking_utility)>
* Wireshark <https://www.wireshark.org/>
* Burpsuite <> Zed Attack Proxy (ZAP) <https://portswigger.net/> <> <https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project>
* NFQUEUE <https://home.regit.org/netfilter-en/using-nfqyeye-and-libnetfilter_queue/>
* PyPacker <https://gitlab.com/mike01/pypacker>
* Scapy and examples <https://scapy.net/> , [https://gist.github.com.eXenon/85a3eab09fefbb3bee5d](https://gist.github.com.exenon/85a3eab09fefbb3bee5d) , <https://github.com/fgrouter/python-netfilterqueue>
* Dr. Acostas terminology <http://www.icir.org/vern/cs161-sp13/notes/Networking-terminology.txt>

# Action Items

This section summarizes the inconsistencies, unclear items, new questions and unanswered questions, as well as next steps that the team has consider as part of the planification of the project. This section, in a sense summarizes our preliminary concerns about the project and lists the elements that in our view were not successfully addressed by the clients during the interview.

## Inconsistencies

* Some network sniffers like tshark, have lately acquired the functionality of intercepting packets, so it seems at first redundant to have two solutions to intercept packets.

## Unclear Items

* Fuzzers: How much flexibility should the analyst have for setting up the fuzzers? Should our system provide details of the fuzzed data to the analyst?
* NFQUEUE: What role would this software play in our system? Is this necessary even if we use other software, for example Wireshark?
* Interception seems to be equivalent to being able to see the packets going through the network.
* Interception is equivalent to getting a copy of every packet? Or is it just being able to observe the network flow?
* Is the client interested in creating new chains of control using IP tables or not?
* What is the format on which packets are saved into the queue?
* Possible scope of refinement of iptables rules, as well as chain rules.

## New Questions

* In the context of the vocabulary used to describe the system is intercepting equivalent to capturing?
* Please provide in detail a realistic testing scenario that could be followed or used to check the reliability of the proxy.
* Is the visibility of packets, like that of “promiscuous mode”, be part of our system, or should the visibility of packets imply interception?
* Should the system automatically transform a packet into a PCAP file once it is received?
* Please describe in a context that includes the network and our system as part of any given network what will be the role of our system, and how can be visually understand the proxy behavior of our system as packets are flowing through the network and our system.

## Unanswered Questions

* Could you please provide a use case scenario on how to set up an iptables rule?
* Can there be multiple [Display]filters?

## Next Steps

* Craft a memo with the inconsistencies, unclear items, new questions, and unanswered questions that came about after discussing the information obtained during the interview.
* Research our questions a little bit further and make more specific records of the things we learn.
* Make models of the system to understand the system better and discover new questions.
* Develop prototypes and preliminary works that can communicate what we understood the project is about and what the needs of the clients are.

# Appendix A

The following is the interview transcript separated by topics. Also included is the presentation before the questions.

## Presentation Transcript:

Elsa Tai- From the University of Texas at El Paso since 2009. He works in the US. Army research lab and leads cyber security research programs related to the development of cyber security decision support, malware analysis, reverse engineering and execution based modern generation graphs. At UTEP Dr. Acosta currently teaches software reverse engineering course. Works closely with the (Door Slam couldn’t transcript). Hosts cyber security workshops at the university. Christian Murga is a software engineering student pursuing his masters of science at the University of Texas at El Paso. As a student researcher he has worked in cooperation with the ROL and the Smithsonian teamwork center to help creating the education video game. In an effort to encourage middle school student minorities to pursue STEM careers. He has worked with the Army research lab creating cyber security curriculum workshops, some of which are in collaboration of the FBI and El Paso electric company. Christian has also worked on developing android applications for the El Paso intelligence center individually. Thank you.

Dr. Jaime C. Acosta - Good morning... so I'm going to start off with a little bit of primer. I know you all got the requirements document, but this will give you a little more context. So we’ll start with primer networking, this is a very high level view but just so you can get the idea of what this project needs and how to move forward. So in this case we have several communicating nodes, the ones that are most important here are the source node, here on the left side, and the destination node here on the right. So, in this case we have a web server, something like google.com or amazon.com. Here on the left side this web client is going to request a web page from this web server and that traffic has to travel through all of these routers in order for that to happen. So internally on each system, on each machine this is the way the network packet is constructed and then deconstructed. Here on the left side we have the web client, in this case and there are several layers here that you can see. So, the very top layer is essentially what pertains to the application, in this case it’s the web server. So, this is where we have the actual request that says, “I need this web page from you”. Each of these layers has a different purpose. I’ll go through those a little bit more in the later slides. What I want you to get from these slides is that each layer, the upper layer uses the bottom layer as sort of a service and same way on the other end when its being deconstructed for reading the lower layer is used by the upper layer. So just to give you a little idea of what goes in each of these layers, like i said this is a very high-level view. The very top layer you usually have your application data, in this case its http but it can be something like an SSH server, skype data. The layer right below that which is called the transport. Is where you have things like, do I need this communication to be controlled, do I want to make sure every packet is received, and I get an acknowledgement for that packet. The layer below that is used for routing, so you saw that it had to travel the packets had to travel through several routers that's what that layer is used for. This is where you will see thing called IP address, Internet protocol addresses. The layer below that has to do with the hardware, the physical layer. So now digging in a little deeper as I mentioned these layers, each of these layers we have things called protocols. So, this is where you can specify how the communication occurs for that particular layer. Here in this case I’m showing you a screenshot of what’s called an internet control message protocol specification. So, these are always in the form of request for comments or RFC’s. You can look these up on the web. These describe very explicitly how that protocol works and what has to go into that protocol for things to communicate properly. Here a visual of what you saw on the previous page. So, these are the definition or the specification for the ICMP protocol. Here were saying that there’s some header data. In that header data you have several fields, and these are the lengths of those fields. Here for example we have an eight-bit field called Version IHL and so on and so forth. So, this is essentially what makes up that ICMP protocol. So, if you were to collect some of this data you could then use some tools, there called network sniffers. In this case your seeing a screenshot of something called Wireshark so you can open up this traffic data and these tools. Then you can analyze what values are at each of these fields. So, these tools usually use something called dissectors, which are essentially parsers that know the specification of the protocol. So, they know how to parse out all the information as you saw on the previous slide for each of those protocols. Here we have a nice display of all of that data. So, what do we need for this class to work on? All the details are in the RDD at a very high level we want to make sure we can intercept packets as they travel through the network. We want to have the ability to modify those packets. So, we might want to change the value of fields, we might want to change the length number. Just to test to make sure on the other end it doesn't break when it gets something that is not valid. After that, after we modify we need to be able to forward it along so it continues to its final destination. there’re several other functionalities that we need that are described in the RDD. So here's the high level logical example of what we need. so, on this side we have a user and again we have a website here. it just switched now its on the left side. So in this case the way that the communication occurs is. It starts like I said before, at the application layer and as the packet gets constructed it goes further down and it keeps getting wrapped in different layers. So, at the very top we essentially have the request that says, “I need a web page”. Right below that we say we want this communication channel to be reliable so we want to get an acknowledgement back for every packet. Below that we have information related to the IP or the Internet Protocol layer. This case we specify a source address and a destination address. So you can see here we’re going from 100020 to 100010. Then the layers below that have to do with the physical connection it could be wireless or ethernet and so forth, So on the other end when it’s received it goes up the stack. So here we dissect the different layers associated with the hardware. Then we extract out the addresses, in this case, this website is going to see the address and say ok this is destined forming, I know who it came from, it goes up the chain. It says ok I need to establish a wireless communication and there are asking for an HTML web page. What we describe earlier is the requirements two slides ago. What we need to be able to do is get in between this communication. So here instead of going to its final destination, we want to be able to intercept this traffic. So some of the things we might want to do in this, during the interception is we might want to change certain fields. In this case we are changing the source IP address. So what that would do is, when it reaches its destination this web server is going to see that it came from this IP address, which in fact it didn't, it came from over here. When it responds it's going to respond to this IP address instead of the original. That's just one example of a field we would want to change. So, if we did this and we analyzed it in Wireshark. So this is a screenshot you saw before. In here it's a little bit hard to see. But you can see some of the fields like, source address, destination address. Here we have some fields that are particular to internet control message protocol ICMP, which we saw on a previous slide. In particular if we wanted to one of the things we would like to do is modify, for example, this type field. Originally it was an 8 which is indicative of a ping request, we might want to see what happens when we inject an FF instead, which is a value that's not defined as part of that protocol. One thing to keep in mind though is that if we changed this, we might want to also change this value called checksum, which is automatically calculated based on the values in the fields for that protocol. So that's what i meant before by upkeep. So these are some links and suggestions going forward we did have some link in the RDD but this are some additional ones that might help you along the way. We want to use a Linux system. It's much easier to interact with the network stack. Kali Linux is a pretty good candidate because of many tools that you’ll need, for example it comes with Wireshark, Burp suite, I think that's it. other would take a longer time to install on the other flavors of Linux are easier to install on Kali. OH! this one is important the second bullet when you are testing you don't necessarily need 3 machines. You can actually do that testing on your own local machine, set up a proxy on your own machine. So essentially your intercepting the packet before it hits your network interface card or physical device in your system. These are some example or helpful links. First one is a link to Wireshark, which is a tool you saw on the previous slide that dissect the packets and lets you see a visual. burp suite and ZAP are web proxies, so they do something similar to what we need except only for the HTTP layer, which is the top layer in the stack. NFQUEUE that lets you intercept packets. PyPacker is a packet manipulation library in python. Scapy is a similar thing except they have trade offs. They do accomplish similar tasks. PyPacker for example is really good when it comes to threading, but Scapy has a much richer framework. So, you might be asking well why do we need this. The main reason is for security testing. So essentially, we need to be able to check for developers that might try to just hide things that are weak in their code. By saying nobody has specification for this so there no way they will know to test this. When somebody just sends an FF for example instead of a 08 or anything that's defined. It might crash the entire protocol and that would essentially execute a denial of service on that system. Another reason is for efficient testing, so we could do some of these things manually, which is the way we do it now. Just takes a really long time. because we have to build it from the ground up. So, the way that we want to accomplish these goals is by attacking or testing against malware protocol we might not have, a specification or maybe its to new. Non-IP protocols so in the example that you saw before with HTTP and even ICMP can be considered IP. They have IP addresses but there is very few software packages that are able to conduct testing below that layer. Others are specification less protocols, so those that don’t have a public specification. There might be some proprietary goes back to the security by obscurity, it might be malware or some other. For fuzz testing, you saw this as one of the requirements. We want to be able to test boundary conditions, unexpected random data as well and also related to fuzz testing speed and loads. So, if the server for example is expecting packets to come at no faster than 1 per second. What happens if we send 2000 in a second?

Elsa Tai- Okay, do you guys have any questions?

Student - So what are the reasons you chose python to do the website?

Dr. Jaime C. Acosta- The website?

Student - Yes

Dr. Jaime C. Acosta - Oh that can be in any language it was just an example that we stole from another slide. But yea that can be any legitimate website as well.

## Q&A Session Transcript

## General Questions Transcript

Student - Could you please describe the different uses of the system.

Dr. Jaime C. Acosta - Sure, I think I described a little bit of it during the presentation but umm, yeah essentially, we see it working in a few different ways. The first way is a live method where we essentially become or we somehow get in between the two communicating nodes and the traffic is live. So we intercept the packets as they are coming into the network live and then we might choose a specific packet and then choose to modify a few fields and then forward it along. The other way we might do it is we load a PCAP file or saved capture and we do something similar. We select the packet we modify it and forward this along.

Student- Another question, could you please describe the current process used to test protocol vulnerabilities.

Dr. Jaime C. Acosta - Yea sure. so usually the way we do it now is mostly manual and the live mechanism where we capture, modify and send is really difficult to do unless we really generate a special purpose script or binary to do that. So it’s mostly manual right now.

## Packet Questions Transcript

Student - What’s your motivation for using AFL for fuzzing when Scapy also support it?

Dr. Jaime C. Acosta - So AFL is a smarter fuzzer in some case there are some trade offs. It’s also what our analysts are more familiar with. Do you want to add anything? At any time don't worry about it (talking to Christian).

Christian Murga - So the thing with Scapy it would know a little bit more about the protocol itself. Where as with AFL you need to specify a little bit more in detail of what needs to be fuzzed. AFL is a little bit more manual but could be better.

Dr. Jaime C. Acosta - It performs better as well. You can actually go to AFL’s site and it has some propaganda there telling you why you should use it.

Student - What can be modified in a packet?

Dr. Jaime C. Acosta - Okay, essentially anything in the packet. So, the way it works is the dissector, so first it is intercepted on the inner network interface card then the operating system deals with certain levels or layers in the packet. Then at that point you can essentially tell it instead of routing it up through whatever processes are waiting for that packet. Or listening for particular things associated with that packet for example port number or other. You want to intercept it and you want to be able to work with the packet. So at that stage you should be able to see everything that’s been dissected. You can leverage things like tshark’s dissectors, Scapy also does dissecting, PyPacker as well. So yeah, anything that can be dissected should be modifiable.

Student - And how should our system handle the packet dissection?

Dr. Jaime C. Acosta - No, you should leverage dissectors that are already implemented.

Student - And how should we handle encrypted packets?

Dr. Acosta - The same way the dissector does, so if it says this is a big chunk of raw data then that’s how you should see it.

Student - How should the syste-

Dr. Jaime C. Acosta - Sorry, I wanted to add one thing to the previous one. One way you could potentially tackle the encryption problem is if you leverage the idea of Hooks. So, if you know about the encryption parameters and you wanna write a hook that will intercept that payload data, decrypt then display the decrypted data.

Elsa Tai - Will the analyst be able to just upload their own dissectors?

Dr. Jaime C. Acosta - Yeah, but in that way, it should be- So just like when you- It should be input into tshark or whatever being to dissect. That’s it’s own engine it knows how to read the dissector. You write in Lua for example tshark is able to load that in and then sees the packet it’ll automatically dissect it using the new dissector. Sorry go ahead.

Christian Murga - You should also be able to load dissectors in order to decode packets

Student - How should the system handle the priority level of the most incoming packets?

Dr. Jaime C. Acosta - Just as they arrive. It’s essentially a queue.

Student - OK, will it be necessary to keep track of the changes done to the packets?

Dr. Jaime C. Acosta - No, but what we do want is a save or a PCAP, a capture for all the packets that were in play and intercepted

Student - Should the system be able to handle other variations of data packets formats?

Dr. Jaime C. Acosta - Yeah, and that goes back to the dissectors so for example if we have proto expert in 1 and proto expert in 2 then if there are differences in the specification of those two protocols. Then they taken care of in a dissector.

Christian Murga - A lot of proprietary protocols where you don’t have the dissectors for. Those are the usually viewed as TCP or UDP. So even though you may not be able to handle the dissector format you should still be able modify the packet

Dr. Jaime C. Acosta - Yeah it won’t burst out the fields, right, it’ll just show it as raw set of bytes

Student - Packet analysis made on the fly using hooks to interpret intercepted packets or will the modification be decided by the user on real time?

Dr. Jaime C. Acosta - Both.

Student - And what will an interactive modifications software packets entail? Does it mean the application should be able to identify the protocol, the available modifications for each protocol, and them prompt the analyst for the desired modification or the software react to each protocol by set of rules.

Dr. Jaime C. Acosta - Kinda like what we described it in the PowerPoint. Um, it’s one of the answers earlier, it can be either live so then you intercept the packet and as it comes in you go and modify the fields then forward it. Or through a previous packet cache or PCAP file.

Student - What options need to be present in the filter that selects which packets are going to be intercepted? What does it mean for a packet to be intercepted? Could there be multiple filters?

Dr. Jaime C. Acosta - Yes, so for filters the way we see it working is first you define a display filter which is a concept that comes from Wireshark and tshark where you essentially say I want to be able to see these certain types of packets are relevant. So, in the ICMP example you would specify ICMP or IP dot proto equals ICMP and that would only bring in filters that is not a ICMP packet. That’s the first stage and then the stage after that is the hook, if you had any loaded it would look at those packets and if there is something that matches the logic in the hook then the rest of the logic would be execute. So for example in this case say the display filter is ICMP and in the Hook file you can say something like if I see ICMP code or dot type. Just one of the fields in the ICMP protocol equals 0008 the you do something with that packet otherwise it wouldn’t.

Student- Uhm I know you already talked about it, but why are you requiring the …. regardings[regarding] dropping are they deleted or are they archived in the history?

Dr. Jaime C. Acosta - No they’re just removed, just dropped.

Student - What conditions do the packets need to meet in order to get captured?

Dr. Jaime C. Acosta - Essentially the filter, that’s what it’s for.

Student - What conditions do intercepted packets need to meet in order to get sent or dropped?

Dr. Jaime C. Acosta - That’s really the analyst’s interaction with the system, so as the analyst clicks through the different widgets in the GUI, modifying fields, there should be something in there that says drop certain packets or forward certain packets or modify and now it’s ready, so now forward the packet.

Christian Murga - To add on, [hooks] should also be able to determine whether it gets sent or not.

Student - When the packet interception mode is on, how should the system handle packets that are previously intercepted and in queue.

Dr. Jaime C. Acosta - So it should just forward everything on that was in the queue.

Student - For packets that are meant to be translated from IPv6 to IPv4, should the system intercept and translate the packets? If so, which tunneling method should the system support?

Dr. Jaime C. Acosta - This goes back to the dissector and the way to operating system handles the packets, so essentially its however the operating system is handling them.

## Hooks Questions Transcript

Student - So I have Hooks for my section. In the RDD it is stated that hooks are functions that execute code snippets and run automatically when some conditions are met. It has a series of questions, so for the first one: What programming languages are the hooks written in?

Dr. Jaime C. Acosta - Hmm, Python

Student - Should the system provide the analyst a way to create hooks on top of loading hooks? If yes please provide a use case scenario on how hooks will be implemented.

Dr. Jaime C. Acosta - Yeah, so this one we thought about it we see, I mean you can write python code in several IDE’s and environments to write python code, so it would not make sense to add that feature into the system, but you should be able edit the files elsewhere and just load them in.

Student - How does the analyst define the conditions that cause the execution of the hooks?

Dr. Jaime C. Acosta - As mentioned previously first through display filter and then through the logic within the hook.

Student - Please provide a use case scenario on how to create and store a hook collection.

Dr. Jaime C. Acosta - So by hook collection we meant several hooks that will pertain to a specific protocol or some type of communication. So, as an example you could set a single hook that is called HTTP. Hook and in there you will specify that you only want to collect or work with packets where the transport layer had a port number 80, so it will be something like TCP.port = 80. So that will capture traditional, non-secure http traffic, you might also define another hook that does something similar except that it is TCP.port = 8080 and another one that says TCP.port = 443 which is https. So, those three hooks can be bundled and into, called a hook collection and you can either offload or load those as a collection.

Christian Murga - You can also bundle them into a collection depending on the purpose not just on what ports are attacking or whatever. So, say a collection of hooks performs a JavaScript injection or some other functionality. So, that’s another reason for collection.

Student - Last question, please provide examples of conditions and statues that can trigger hooks.

Dr. Jaime C. Acosta - I think we gave some examples, right? TCP.port = 80 TCP.port = 8080 and it doesn’t have to port right it can be any field in a packet.

## Protocols Questions Transcript

Student - So, onto the next section on protocols. So, you mentioned that Burp Suite is a software system that kind of performs a similar task but only for just http protocol. You have already previously stated that [Inaudible] should the system support any type of protocols including proprietary protocols, and you said of course yes.

Dr. Jaime C. Acosta - Yeah.

Student - And then, a quick follow up. Will those protocols have to defined and loaded from external files by the analyst or are they need to be able to be defined from the program itself?

Dr. Jaime C. Acosta - So, now those will be loaded and dissected according to the dissector that is available.

## Iptables Questions Transcript

Student- Please provide a use case scenario on how to setup an ip-table.

Dr. Jaime C. Acosta - So, in the RDD I provided a link to an example on how to use Scapy and NFQUEUE, so in that example if you look at the source code there a few iptables rules that are imbedded into the script. Some of the links I provided here have the same sort of examples.

Student - Ok, the second question for you is what sort of filters would you like to specify to determine which packets are intercepted?

Dr. Jaime C. Acosta - Display filters.

Christians Murga - Sometimes for those is BNF.

Student - Could you repeat what you said sir?

Christian Murga - Oh, the syntax for the filters should be in BNF.

Dr. Jaime C. Acosta - So, the way we kind of see it working is you specify the filter so and after you do that, so essentially that is what is only going to be captured. After that you will hit the intercept button and that is when you will actually intercept those packets and pause them before they get forwarded along. So they are two separate things.

Student - Onto the next question, please provide a use case scenario on how to configure a filter to intercept live traffic.

Dr. Jaime C. Acosta - So, I guess at a very high-level user starts the system, user enters a display filter, and user clicks on the intercept button. And I guess that you will have to specify the interface in there too.

Christian Murga - Optional.

Dr. Jaime C. Acosta - Or, if you don’t it will just capture on all interfaces.

Student - And our last question from iptables, the system requires analysts to set rules for iptables and save configurations, but should we anticipate the possible use of ip6tables(IPv6), ebtables(Ethernet Frames), arptables(ARP)?

Dr. Jaime C. Acosta - That was a good question. For IPv6 don’t worry about it yet, but scope the software so that it will scale, we do want that capability. One thing that we have run into while we have been researching this and testing things. Iptables works well for ip layer and above but ebtables may be a solution for the layers below that, another solution might be docker containers or some sort of virtualization.

## Fuzzers Questions Transcript

Student - Onto the next section, the next section is fuzzers, should the project be able to fuzz packets on the fly or is fuzzing expected to be a process of its own?

Dr. Jaime C. Acosta - It is a process of its own but the way to configure things for the fuzzer should be within the system and that could be done in a live fashion. But the fuzzer on its own will run on a different process.

Christian Murga - If I could add onto that, in other words anytime a packet comes into the system either through a live tracker or through a PCAP you should be able to send any of those packets to the fuzzer.

Student- Should the packets generated by the fuzzer count towards the queue of packets or is the fuzzing process separate from the other processes?

Dr. Jaime C. Acosta - Separate.

Student - Since we are trying to fuzz packets and these packets might be completely unrelated to one another, in what way would the fuzzer test its mutations? If the mutations are not to be tested should the generated mutated packages be saved in a PCAP file for individual evaluation?

Dr. Jaime C. Acosta - Yeah, save them to a PCAP file, that way we can always go back and see what was actually executed, what might have been missed. Yeah, that’s a good question.

## Users Questions Transcript

Student - Um okay, so umm just a few questions regarding people who will be using this system, it’s mentioned in the RDD that umm the system should be used by intermediate to expert level analysts, right? Umm could you possibly describe the difference between these two types of analysts and how they might use this system?

Dr. Jaime C. Acosta - Sure, so in the intermediate, the intermediate and above we see someone that’s familiar with networking technologies, umm they have a understanding of how nodes communicate, but what separates the advanced from the intermediate is that the advanced would know, for example, how dissectors work, they’ve written a few of their own, they know about specifics about the protocol and what’s required for the connection to take place, for example, TCP which is um a standard, has to establish a handshake which is a three way communication before communication can occur, so something like that or other protocols would be able to separate an advanced user from an intermediate, someone that can write hooks, because they have that knowledge would be more of an advanced user.

Student - Okay, and one more question.

Student - Do you want us to generate logs for usage of the system, for example if someone uses the system, do you want to generate a log showing the activity?

Dr. Jaime C. Acosta - That would be nice, but it’s not a hard requirement.

## Display/Interface Questions Transcript

Student - Okay, I got the display interface the first question is what does the nodes are needed besides raw hexadecimal and the code type?

Dr. Jaime C. Acosta - Also a binary view, so sometimes there are fields that are smaller than a single byte, so it might be a 4-bit field, so we should be able to interact with that as well.

Student - Okay, the second question is, what does it mean for a type to be the code and how the system should handle a type decoding failure?

Dr. Jaime C. Acosta - So the type is kind of a artifact of a dissector, so a dissector knows how to parse out the fields in a packet so if it knows, for example, the third field in this particular packet and it’s 4 bits long and that pertains to a integer, then it would parse it out as a integer type, that's what it means by that. But for some reason that decoding doesn't work, the user should be notified somehow, we don’t want the system to crash, but we want at least just some notification.

Student - The third question is how should a queue of intercepted package be displayed on a GUI?

Dr. Jaime C. Acosta - So we should leave this one up to the developers, we have some ideas, but we want to see what you guys come up with for this one.

Student - And the last question, are there any GUI elements that exist in other tools that we should consider when designing the system?

Dr. Jaime C. Acosta - Yea, so I highly recommend looking at some of the software that I left some links in the PowerPoint, we’ll provide that after this interview, Wireshark is good, Burp Suite and zap, probably the most relevant.

## Security Questions Transcript

Student - Um, I’ll ask you some security questions. Do we need to take into account any special security features on what the software such as needing to work on both contractor and civilian computers, handling classified or sensitive information, or running in labs with an over the air barrier (no internet access of any kind, and no outside software allowed).

Dr. Jaime C. Acosta- No that will all be handled in a different abstraction than within the system. One thing we do want to mention though is if in the case you, maybe one of the dissectors, or something you are working has to execute code that depends on the packet or some data in the packet just make sure you execute it as a non-trusted user, otherwise you might exploit or be victim of an exploit or a vulnerability within the system.

Student- Having said that should the system support user authentication at all?

Dr. Jaime C. Acosta- No, [Inaudible]

Student- Final question, should data generated by the system be encrypted?

Dr. Jaime C. Acosta- No.

## Development Constraints Questions Transcript

Student- Ok, for the last section on development constraints, what are the constraints in regards to programming languages, such as the version and GUI libraries? I have here that for [Inaudible] Scapy supports both python2 and python3. Python2 will not be supported beyond January 1st, 2020. Is it acceptable to use python3 for the application?

Dr. Jaime C. Acosta - Yeah, Python 3 should be, it is fine.

Student - And then, in the RDD, it is stated that the system should support queues of at least 100 packets without any behavioral performance issues on a modern laptop system. Please define what constitutes a modern laptop and any performance requirements.

Dr. Jaime C. Acosta - Ok, so well 7th generation core i5 with 8gb of RAM.

Student - What operating systems should the system support?

Dr. Jaime C. Acosta - This was mentioned before, I think kali it is probably a good choice, let’s go with Kali 2018.2, since that is the most recent version.

Student - [Inaudible], Scapy has a GPL license. That means, in effect, anyone with an executable of the application needs to have access to its source code, including other DoD organizations and contracting firms. This also means that any derivatives of this software need to have a GPL license. Is that acceptable?

Dr. Jaime C. Acosta - Yes, that is acceptable.

Student - And then, for sure last question, for hooks, it can be fulfilled using Amazon’s AWS Lambda resource or Microsoft’s Azure functions. Would you be open to using Lambda or Azure for this system?

Dr. Jaime C. Acosta - No, not for this project it introduces too much overhead and costs. One thing I did want to clear up, I mean I heard a lot of Scapy and Scapy could be the solution but also look at PyPacker and other options as well.

## Extra Questions/Follow-Ups Transcript

Elsa Tai - Does the class have any follow up questions for constraints?

Student - For the first question in development constraints did you answer that or?

Dr. Jaime C. Acosta - Yea Python 3. Is that the question you are referring to?

Student - Yes

Student - I was going to re ask you, with a question in the same section. What operating system should the system support? All of them right?

Dr. Jaime C. Acosta - No, at least Kali 2018.2

Student - So you have as of now no intention to extend this to any other, more common operating systems at the time.

Dr. Jaime C. Acosta - Linux in general

Student - ok

Student - In your PowerPoint you said it was unclassified. Does that mean that this part of it could be open source?

Dr. Jaime C. Acosta - Sometime in the later future yea

Student - So should we build it with that in mind or should we start?

Dr. Jaime C. Acosta - yea so are you asking if you could bring in proprietary software and things that cost money that can’t be made open source later?

Student - sure

Dr. Jaime C. Acosta - Ok, then yea that would be preferred. Although that’s open for discussion.

Student - Christian mentioned that the filters should be in BNF format can you clarify on that?

Christian Murga - Inaudible

Dr. Jaime C. Acosta - it’s a standard format, we would like to continue using that. We will provide that along with the slides.

Elsa- Any other questions for our clients?

Student - Would the PCAP files be stored on the local machine, or would the user be able to download those from the system?

Dr. Jaime C. Acosta - They should be stored on the file system itself, but somewhere in there through the interface it will specify the default directory. Question back there?

Student - Same question as before except for the hooks

Dr. Jaime C. Acosta - yea somewhere on the local file system, again configurable on the GUI default directory

Student - So where would this program, I know you mentioned you can run everything off your own machine. but in an actual testing and everything, would it be running on the sender machine, receiving machine, or somewhere in between?

Dr. Jaime C. Acosta - It can be any of those. So, you can set it up to run on either the machine the client, for example or you can set it up on the web server part. Or if you somehow become a monkey in the middle, it’s what it’s called, then you could get both sides.

Christian Murga - Actually doing this kind of development and testing is done on virtual machine so it doesn’t mess up your iptables and system.

Dr. Jaime C. Acosta - That’s true good point. Virtual machines. Good question.

Student - So the hook system you did say that you wanted to interact with a display filter. so does that mean you only want to interact with things you filter out, or should our filter be triggered by anything or anything in our system?

Dr. Jaime C. Acosta - No its only what, based on the filter, that’s what the hooks would see. Otherwise the performance would suffer.

Elsa Tai - any other questions? No? Let’s thank our client.

∎

**Interview Report: Grading Criteria**

**Team 5: 75**

|  |  |
| --- | --- |
| **50 pts.** | 1. **Complete**  * All required sections are present: * Document Control (1) -0.5 * Title Page (1) * Introduction – Purpose (16) * Purpose of this document, what the intentions are, and who are the primary intended targets (5) -0.5 * Project overview (5) * Background and contact information of interviewee (4) -1 * Interview information (2) -0.5 * Questions and Responses (12) - 3 * Action Items (10) * Inconsistencies -2 * Items that remain unclear -2 * New or unanswered questions -2 * Next steps -1 * Sufficient details are given to show understanding of interview. (10) -3 |
| **20 pts.** | 1. **Correct**  * Information presented is correct * Information is properly referenced -1 |
| **15 pts** | 1. **Consistent** -0.5 |
| **15 pts.** | 1. **Presentation**  * Clear and concise -1 * Correct grammar -1 * Professional presentation -5 |