LAB 3 COVERT COMMUNICATIONS

Dennis Keritsis

Steps 4, 4a, and 5

Requirements

1. Hide a basic message of "Hey, this is a secret message that totally contains secret stuff" into the image (originally a .jpg) provided by Instructor.
2. Hide another image within the Instructor provided image (.jpg).
3. Investigate whether secret information (e.g. another image or secret messages) visually taint the physical image, and at what point.
4. Investigate properties of files (e.g. hash values and file size).

Reference Table

I will discuss findings in later sections. I want to layout a table such that you can clearly reference all my data and not become discombobulated with the glut of the file submission.

Name of Picture or Data Set	**Type**	**-ef or -cf**	**Script Associated**
Under Images Directory			
Jason_Basic_message	.jpg	-cf	jason_basic_1
JasonMessageReplete_run3	.jpg	-cf	how_large_3
wmuhorzgold	.jpg	-cf	N/A (original submission)
wmuhorzgold_after_alot_of_ones	.jpg	-cf	how_large
wmuhorzgold_after_full_lamb	.jpg	-cf	how_large_2
wmuhorzgold_Embedded_with_google_BMP	.jpg	-cf	jason_basic_2_image_in_image
Google_other_image_BMP_version	.bmp	-ef	jason_basic_2_image_in_image
Under Misc. Directory			
correlation_between_secret_versus_image[1]	.txt	N/A	correlation_of_how_large
hash_original.txt	.txt	N/A	Created form command line: sha1sum wmuhorzgold
secret_after	.txt.	N/A	File created from the bash scripts during runs
secret_before	.txt	N/A	File created from the bash scripts during runs
Notes.txt after %	.txt	N/A	Personal Notes during runs

Requirements 1, 3, and 4

Foreword

[1] This is experimental data. See discussion on this below and excel file.

I will discuss requirements 1, 3, and 4 together. The discussions below are grouped in terms of "script runs" (as I will call them). Requirements 1, 3, and 4 will be discussed accordingly.

Script Run —jason_basic_1

This was relatively straight forward. I ran my "jason_basic_1" script which utilized, among other things, the following command:

steghide embed -cf wmuhorzgold.jpg -ef secret_before.txt -p hello

This command utilizes the switch -cf that specifies the "cover file" (i.e. the file to be embedded), and utilizes the switch of -ef that species the "embedded file" (i.e. the file inside the file that is meant to be secret). Optionally, you can encrypt with the -p switch. In this case, I specified the passphrase of "hello." There was no visual difference to the naked eye. (Appendix A.) The hash of the original file before embedding was: e4ccb4dafa89b50ffae7a398b6413b793fcbef2b. After embedding the hash (to no surprised) changed. Hashing is the best form of data integrity. The original file had 71,738 bytes of data. After embedding, the new file had 71,779 bytes of data. This can be viewed as a more elementary form of data integrity.

Script Run —how_large

I ran some preliminary scripts for experimentation purposes. I had a simple string of length one (1) with the number (1). Then I ran a while (TRUE) loop too see how fast the file size would grow and how large the file size could get.[2] The file to be examined is wmuhorzgold_after_alot_of_ones. If found that the picture quality was tainted but not too bad. (Appendix B.) This makes logical sense since as the embedded file data grows (even with compression) the cover file will be more tainted as a result and it will be harder to "hide" data to the naked.

Also noteworthy, the hash value was changed. The secret was 113,590 bytes, according to my notes. At this point, I came to the conclusion that there must be some type of compression algorithm at work since the secret size was larger than the cover file (i.e. 113,590 compared to 72,251).[3]

Script Run —how_large_2

Further engaging in experimentation, I ran a script with the same style TRUE loop with the string "Mary Had a Little Lamb, and her fleece was white as snow." As to be expected, the file growth was at a faster ran since the string was larger. This time I was able to get the embedded file to a size of around 78,174 compared to the original of 72,251. Finally, during my run I found the following returned message:

[2] I could just break out of the TRUE loop with Ctl+C. In fact, I did this for a lot of the bash scripts cause I n00b.
[3] See my Notes.txt after % for further details.

"cover file is too short to embed the data."

At this point the secret and/or the -ef file was too large compared to the cover file. I also took note of the correlation between secret size and the embedded image. I also took note of the file's image quality. It was clearly tainted. (Appendix C.)

Script Run —how_large_3

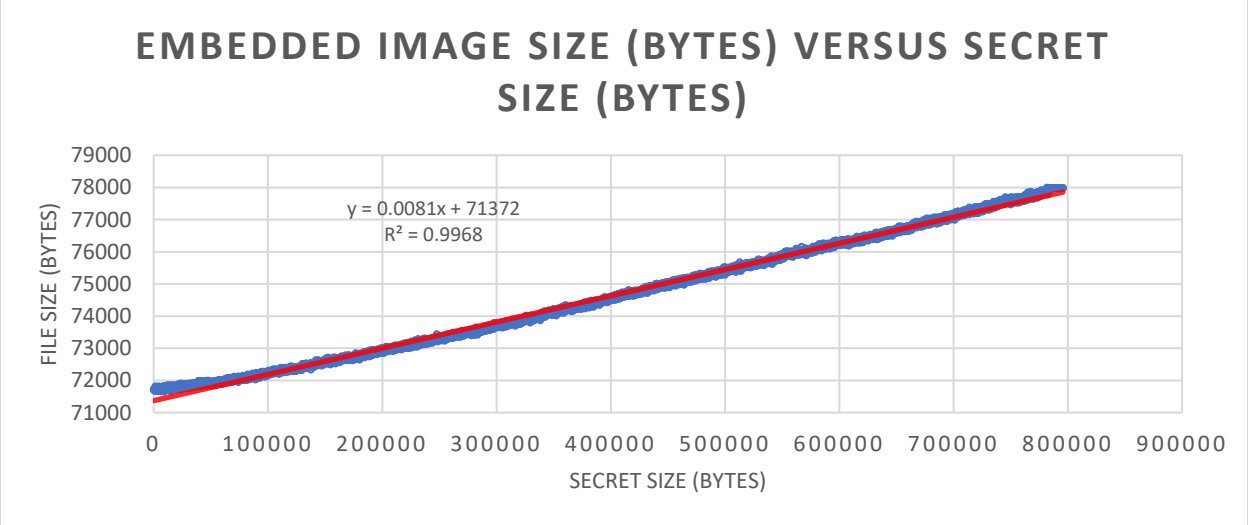
Further engaging in experimentation, I ran a script with the same type TRUE loop with the string: "Hey, this is a secret message that totally contains secret stuff." I wanted to embed the file with a difference string to see any empirically related differences when compared against the "how_large_2" script. There was nothing readily observable. Again, the file image of the cover file was visibly tainted. (Appendix D.)

This was possibly due to the string being of similar character as the "how_large_2" string. Further experimentation is warranted.

Further research should include running a series of scripts with **different** informational entropy to see how the compression algorithm would handle this. Also running scripts with **different** lengths, possibly up to the thousands to see how the compression algorithm would handle this.

Script Run —correlation_of_how_large

After experimenting I ran a script to gather data that compared the secret time to the image size and appended my results to a file.[4] There is a strong linear relationship (i.e. R^2=0.9968) of file size (cover file) to secret size.



[4] Please see correlation_between_secret_versus_image for RAW data.

As I mentioned before, further experimentation is warranted. It would be interesting to see how this relationship changes *vel non* based on string length, string entropy, and other string parameters.

Looking carefully at the excel file or the RAW data,[5] the max file size for a resulting file is 77989 bytes. This corresponds to a secret size to about 783156 bytes.

Script Run Synopsis in Table

Image Name	Embedded File Size (bytes)	Appendix	Visual Note
Jason_Basic_message	71779	A	No visual diff
wmuhorzgold_after_alot_of_ones	73326	B	Ok
wmuhorzgold_after_full_lamb	78174	C	Bad
JasonMessageReplete_run3	78320	D	Bad

There is a relationship between visual efficiency and the amount of data embedded into the cover file.

Requirement 2

Script Run —jason_basic_2_image_in_image

I noted that there are requirements for the embedding file (-ef) when it comes embedding an image with an image. Users can only embed JPEG's or BMP. Apparently, the file type of the cover file matters not. Jason's original cover file was JPEG file, which is not supported of the -ef file.

Originally, I tried to embed a JPEG into the JPEG (WMU). However, the cover file was not large enough when compared to the -ef. (Please note the excel picture above that notes the max secret size is about 77,989.) As a result, I converted the -ef file to a BMP file since it takes up less space. The BMP file was 1542 bytes whereas the JPEG of the same quality was 19,673. *See* Appendix E for the Thumbnail of the BMP image.

After embedding I found no visual different to the naked eye. (Appendix F.) As always, the resulting hash differs.

Step 6(a) Physical Media

Discuss policy and technical options for mitigating information transfer using physical media.

Overview

[5] Excel file is in the submission or you can view the RAW data.

People are the greatest security risk. This section will discuss vulnerabilities associated with physical media. However, it must be noted that people are at the root of the people, and that the physical media is only a tool.

One type of physical media is the USB. When compared against other types of physical media, USBs have more mobility, are more pervasive, are more convenient. (Liao p. 1.) As such, USBs should be classified as high risk. (Liao p. 1.) Additionally, some media can hold up 100GBs (Heikkila p. 22) and the transfer of data can be considered frictionless in that the transfer of many valuable files can occur very quickly. Put another way, long gone are the days were people would steal mountains PII from paper documents in a picked locked file cabinet.

Some value information includes, personal information, financial information, network information, and illicit material (which is none exists in this paper). (Chaerani p. 33) As such, this not only puts individuals at risk but also business and governmental entities at the following rates 50% commercial, 31% educational, 18% governmental. (Chaerani p. 33) Technical and policy remedies will be discussed. The emphasis will be placed on policy measures.

Technical options — USB Authentication and USB Partitioning

Portable media is classified as high risk. Even if this is the case, many IDS's may not detect the use of portable media given their trusted nature on the network. (Heikkila p. 22) As such, it is important to have a technical authentication mechanism in place to disallow any user from using a specific portable media on the network.

The technical aspect of USB flash drive authentication re described in Jeong. Jeong discloses the use of "hash" function (or alternative encryption functions) to secure the flash drive.

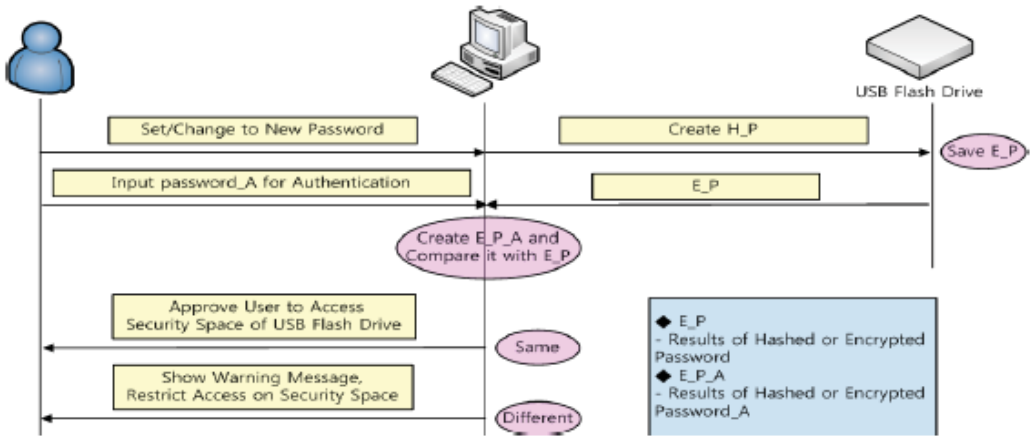


Figure 8. Security Program using Hash or Encryption Function

Figure 1 reproduced from Jeong et al.

This is one embodiment within the instant disclosure. Additionally, Jeong discloses the "partitioning" of the drive into a secure part and unsecure part. (Jeong p. 61.) As such, if the user wishes to access the secure part of the drive, they must authenticate themselves.

It is important to note that the state of the art of encryption may involve the utilization of software-based techniques, hardware-based techniques, or both. (Liao p. 2.) Generally speaking, software has less performance and must be installed whereas hardware is more expensive but has more performance. (Liao p. 2.) As disclosed in Liao, It is important to note that ARM based controllers do exist for hardware-based encryption. (Liao p. 6.) Specifically, Liao discloses the use of chaos authentication to create a secure USB flash drive. (Liao p. 2.)

What should be used is up to an individual and/or the organization because each work environment or user environment has different security level requirements.

Policy options — Please Pick me Up

While it is important to have secure technical equipment, the equipment will be rendered useless if the individuals using the equipment are not properly trained and aware of security risks. According to Charani (p. 33), only 10% of companies use encryption and only 21% of employee recognize USB media as a risk.

This risk is pervasive. A study of "physical trojans" was conducted by Ticher et al (p. 306). They placed an envelopment in an open environment and labeled it with a sticker call "PRIVATE." (Ticher p. 306.) 45% to 95% of users picked up the envelopment, gathered the USB into, and plugged the USB into the computer accordingly. (Ticher p. 306.) 68% took no precautions and 16% scanned for viruses after picking it up. (Ticher p. 306.) A mere 8% just trusted their system to take care of any security problems. (Ticher p. 306.) The authors concluded that this was an effective social engineering mechanism given that people "altruistic" or just curious. (Ticher p. 306.)

As such, it is important to have training sessions, regular audits, updated passwords every 90 days, and lockout policies in place. (Heikkila p. 23.) Additionally, a formal notification mechanism should be in place such as the signing of a document state they understand the media security policy and will abide by its terms. (Heikkila p. 23.)

Having strong policy measure in place with allow employees to utilizes technical devices more effectively.

Step 6(b) Trojans

Choose a second risk: Trojans.

Discuss policy and technical options for mitigating the second risk.

Overview

The name Trojan is taken from the Greek myth as this malware is typically wrapped into a over executable with an underlying covert and malicious executable. (Mell p. 2-4.) This type of malware can pose risk to privacy, money, devices, and file integrity (Saracino p. 83), while also causing wide spread damage. (Mell p. ES-1.) Further, malware is only on the increase. (Saracino p. 84.)

Mobile devices are also in the increase, and Trojans are increasingly being directed towards mobile devices. (Saracino p. 83.) There are at least 1 million mal apps in the "wild" today. (Saracino p. 83.) Of the malware out, 98.5% of attacks are directed towards Android even though Android OS makes up 80% of the market share. (Saracino p. 83.)

Technical Aspects —Signature and Anomaly Techniques

Trojans are executables. They may intend to do a number of things such as: create backdoors and steal cookies. (Mell p. 2-6.) Backdoors, for example, may ultimately be used to create zombies or install remote administration tools (RATs). (Mell p. 2-7.) Ultimately, these may be used to transferer files, acquire passwords, or executable commands. (Mell. p. 2-7.)

From my experience, a type of backdoor may open up a shell in the background of the victim computer and "shovel" it to a hacker client. This can be done through the use of Netcat being install on the victim computer and opening up a listening port. They have a listening component may be TCP or UDP. (Mell p. 2-7.) Ultimately, once the hacker has the shell of the victim computer, they essentially own the machine.

However, there are a number of ways to mitigate these threats. Given the increasing presence of Trojans, there are a number of technical techniques that may be used to detect Trojans, namely, (i)signature-based techniques and (ii) anomaly-based techniques. (Skukla para. 0008) Signature method techniques are considered by some to be "traditional" in that they use name of file, checksums, strings, and the like. (Skukla para. 0078.) However, these methods have limitations given that some signatures are not widely registered to the respective "new malware." (Skukla para. 0078.)

In order to overtime these limitations, it may be recommended to additionally use anomaly-based techniques. (Skukla para. 0079.) Specifically, the analysis of "behavior" of API calls. Malware must use API functions calls, and therefore, it is very hard for malware to circumvent this type of API function all analysis. (Skukla para. 0079.) Additionally, Trojans may file themselves into the registry and process therein. (Skukla para. 0080.) Also, malware may be in a driver. (Skukla para. 0084.) These types of Trojans may be difficult to detect. (Skukla para. 0084.) However, while using analogy-based approaches, this may help to mitigate some of this risk. (Skukla para. 0084.)

Specifically, there is an anomaly-based methodology (Saracino p. 83) called MADAM which operates on "four levels" as follows: kernel, application, user, and package. (Saracino p. 83.) MADAM uses an "anomaly-based approach" and looks at the patterns exhibited by the malware. (Saracino p. 83.)

Policy Aspects

As noted above, the most important aspect of security is its people. According to NIST, it is important to have training and make users aware of the risks. (Mell p. ES-2.) NIST outlines the lifecycle of malware as follows: Preparation, Detection and Analysis, Containment, Eradication, Recovery, and Post Incident Activity. (Mell p. ES-2 to ES-4.)

Trojans may commonly be found at hotels, coffee shops, and external locations on unsecure networks. (Mell p. 3-2.) It is important to have policies that prohibit the use of business computers on these types of networks. (Mell p. 3-1.) Additionally, there need to be training on not opening types of emails, clicking popups, and visiting websites that are known to be harboring malware. (Mell p. 3-1.) Additionally, computer should not download, click, or open filetypes that are known to harbor malware such as: .bat, .com, .exe,.pif, and .vbs. (Mell p. 3-2.)

Additionally, when a user is on the network, it is important that media plugged into the network is scanned. (Mell p. 3-1.) Also, if there any attachments on email these should be scanned too. (Mell p. 3-1.)The busines network use have proper software updates, firewall configurations, restricted USB usable, and restrict use of mobile devices. (Mell p. 3-1.)

Step 6(c) SMTP with Covert Comm.

Q1. How could two people (or more) use an online email service (Gmail, Yahoo, etc.) to communicate covertly without ever sending an actual email?

A1. The paper will discuss remoting into a STMP server, and will discuss embedding information into STMP headers. See discussion below.

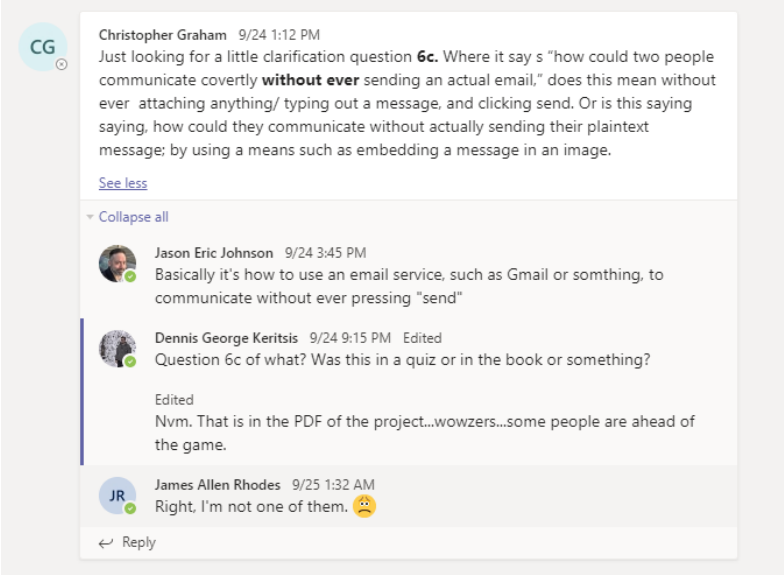
Q2. What options are there for mitigating this risk?

A2. Paper will discuss the use of interposing a specific hardware element, a router, between the STMP server and the SMTP client. See discussion below.

Q3. Which service would you choose and why?

A2. STMP will be the protocol of choice. It is the nature choice since we are sending emails, and the protocol headers give the user the ability to embed a large payload. See discussion below.

Consider things like anonymity of the account owner, how the service archives traffic, emails, and drafts, security of the network traffic while composing messages, etc.



Per the chat, the paper will discuss sending a message without pressing the "send" button but will discuss the use of SMTP servers and SMTP clients to send emails.

Overview of Covert Channels

The network protocol stack, as is well known, has at least five layers. Alternatively, it is well known that the OSI model has at least seven layers. It is also well known which the protocol stack models, there are encapsulation and decapsulation. This involves adding or removing layers when traveling down or up, respectively, the protocol stack.

By this very fact, network protocol provider for a large number of covert channels to be embedded within each of the number of layers with the protocol stack. (Zander p. 44.) This discussion will use the preferred language of covert channel; however, words such as hidden information, covert information, information hiding, and stenography are used within the art. (Zander p. 45.)

In essence, covert channels are simply secret channels within an overt channel. Analogous to Trojan, which has an overt program and a covert program, the covert channel is embedded into the overt channel, wherein the overt channel would correspond to anyone of the number of protocols within the network protocol stack.

For example, Loki is a well-known program for covert channel creation. It primarily uses, among others, ICMP as the overt protocol of choice. (Glotz p. 6.) Within this over protocol, it may embed a number of secret messages. (Glotz p. 6.) These secret messages may include malware or requirement such as: spam, worms, and unauthorized communications. (Rand para. 0077.)

As such, it is clearly that covert channels may be used to transmit secret information as the data is embedded accordingly. This type of communication may not be permitted by an organization. Also, it is important to note that the "secret" information of the covert channel is

different from that of encryption. (Castiglione p. 503.) Encryption provides confidentiality and does not provide secrecy. (Castiglione p. 503.) Covert channels conversely provide secrecy but not confidentiality. (Castiglione p. 503.) However, as discussed above in the stenography examples of the JPGs, one skilled in the art would appreciate, using the -p switch, that the compressed file may additionally be associated with a cryptographic key. This may be called a stegokey. (Castiglione p. 504.)

As a background, STMP is a "text-based protocol" and utilizes a TCP connection. (Castiglione p. 505.) Commands are typically initiation by the client. (Castiglione p. 506.) Additionally, there may be four entities running: email client, sender side MTA, recipient side MTA, and receiver client. (Castiglione p. 505.)

How to send email

How to send email —Commands

As discussed above, the paper will discuss the use of the STMP protocol. However, it is important to discuss the structure of the network when using the STMP protocol. As discussed above, the network may have an email client, a sender side MTA, a recipient side MTA, and a receiver client. (Castiglione p. 505.)

MTAs "spool" the email and look at headers to "find-out the recipient" as indicated by the "To:" header. (Castiglione p. 505.) DNS from the client MTA is queried and the resolved IP address is sent to the recipient MTA with the STMP protocol. (Castiglione p. 505.)

The following commands/steps below discuss the operations that are to be performed by the client while remoted onto the STMP Server (also known as client MTA). (Valentino p. 1 of 1) For brevity, the paper will post the commands as see below:

- On our client we want to telnet to the "STMP Server" using port 25 which is the standard port for STMP protocol.
- WE "greet" the server with HELO command.
- We wait for a response back from the server with a 250 reply.
- We now define the MAIL FROM:
- Wait for 250 rply.
- Define the RCPT TO:
- Wait for 250 reply
- Next we can type DATA to create the subject of the email
- Set SUBJECT: and type up the body of the message.
- We need to put a dot (.) at the end and hit enter to sent the message.
- Then "QUIT" to exit out of telnet.

(Valentino p. 1 of 1.)

Following these steps above would allow a client to use the STMP server to send an email while utilizing port 25 of the STMP server with telnet. (Valentino p. 1 of 1.) When finished, just type QUIT to end the remote connection of the client to the server. (Valentino p. 1 of 1.)

How to send email — Tunneling/Embedding

The previous section discussed a series of pedestrian steps to send an email with "pressing a button." However, this paper is about covert channels. And it would be no fun just to send an email without any legerdemain. As discussed in the overview, protocols like ICMP are perfect for embedded data. (Goltz pp. 6-7.) Similarly, this section will discuss the use of STMP to have data embedded in the headers accordingly.

Given the way SMTP is structure as a protocol, STMP headers that are "manipulated" is not regarded as suspicious. (Castiglione p. 505-506.) Some of these fields include the "received" field and the "id" string, which are both free format ASCII. (Castiglione p. 505-506.)

Additionally, the "Message – ID:" field is an identifier which refers to a version. This is only machine readable and not to be used by humans. (Castiglione p. 505-506.) Typically this is generated by the host and will vary among software products. (Castiglione p. 505-506.) Also, there is no limit to the message ID. (Castiglione p. 505-506.) But it is important not to make it too lengthy to raise suspicions. (Castiglione p. 505-506.) These are the primary headers to have data embedded in accordingly. However, if we need more "bandwidth" we can also modify the "in – reply – to" and "references" headers. (Castiglione p. 505-506.)

Also the STMP protocol utilizes MIME and the subparts of it has the "Content—Type" header. (Castiglione p. 506.) The body of the email is also defined using the MIME standard. (Castiglione p. 506.) These may be used as a vehicle to have data embedded therein, along with the "boundary" parameter. (Castiglione p. 506.)

It is important to note that using a single email with header is not the only embodiments. As noted above, users can increase bandwidth by utilizes more headers. (Castiglione p. 506.) However, users are not limited to a single email. (Castiglione p. 506, 508.) Users may create spam like emails for communications. (Castiglione p. 506, 508.) Therefore, alternatively, a user may "spam" emails with embedded headers to increase bandwidth. Also, spam is replete in today's society and an interceptor viewing the emails may disregard them as trash even though they have embedded secret data. (Castiglione p. 506, 508.) Further , the spam email adds a layer of obscurity to the process. (Castiglione p. 506, 508.) Further still, this would increase the amount of "noise" on the network because user would have the options to use less headers in each of the emails. (Castiglione p. 506, 508.)

How to send email —Encryption with Tunnels/Embedding

The discussion thus far about STMP, STMP embedding (tunneling), and the additional use of spam only contributes to obscurity. As is well known is the art of information security, security through security will ultimately fail. Additionally, obscurity is not part of the well-known CIA triad of Confidentiality, Integrity, and Availably. Tunneling and embedding do not provide confidentiality. They provide secrecy.

However, as discuss in the stenography picture embedding examples, users may embed data with the -p switch.[6] Therefore, the covert channel may also be associated with a cryptographic key. (Castiglione p. 508.)

There is an underlining methodology to this described by Castiglione with his use of a "stego-key." (Castiglione p. 508.)

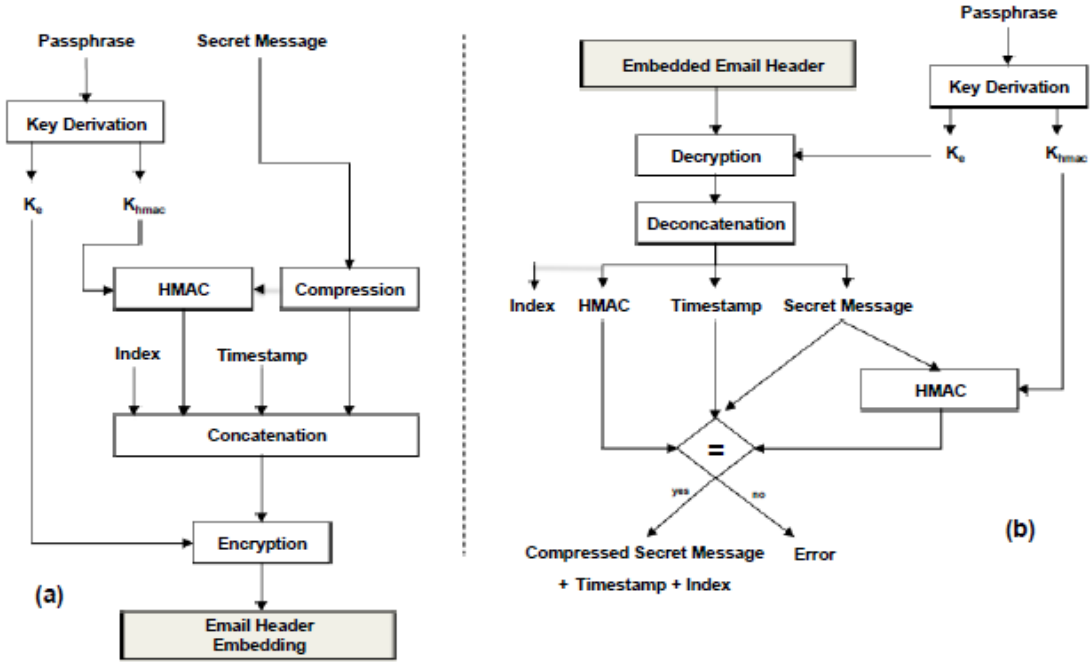


Figure 2 reproduced from Castiglione.

Castiglione discusses the use of encryption (for confidentiality), compression (for saving space), and embedding (for secrecy). Additionally, an HMAC is used, which is a keyed hash value, in order to provide integrity. As such, it is appreciated by one skilled in the art that covert channels can additionally offer confidentiality and integrity from the CIA triad.

Technical mitigation of Covert Channels

[6] Above, I used a simple passphrase of "hello" with -p.

Technical Mitigation — Technical background

Now that covert channels with STMP have been discussed, this paper will discuss ways to mitigate the use of covert channels later. However, a back will be discussed.

Firewalls are a well-known tool to block traffic. (Rand para. 0007.) For example, ICMP can have a covert channeled embedded therein. (Zander p. 50.)However, it would be trivial to one skilled in the art to block the ICMP protocol, and therefore, remove this attack vector.

The example command above which discussed a telnet option to send an email are to be performed remotely by a client on an MTA server. (Valentino p. 1 of 1.) However, the use of MTA intermediary hardening may not be allowed by an ISP. (Rand para. 0084.) For example, if we want to mitigate the risk of remote control of the MTA through, it would be trivial to close port 25 and disallow its use, along with the closing of other ports.

But again, the ISP may not allow this. (Ran para. 0084.) Therefore, on the client side which may be a business network, there must be some other type of solution.

There is a need for an improvement to the state of the art. The interposing of a route with a policy table will be discussed.

Technical Mitigation — Router as Hardware "Customer Facing Router"

Central to the inventive concept is the "transparent relay 710" which "operates discretely and transparently" without the reconfiguration of the email client or the MTA. (Rand para. 0085.) Therefore, this offers a level of portability and flexibility amongst a plurality of MTAs and email clients. (Rand para. 0085.)

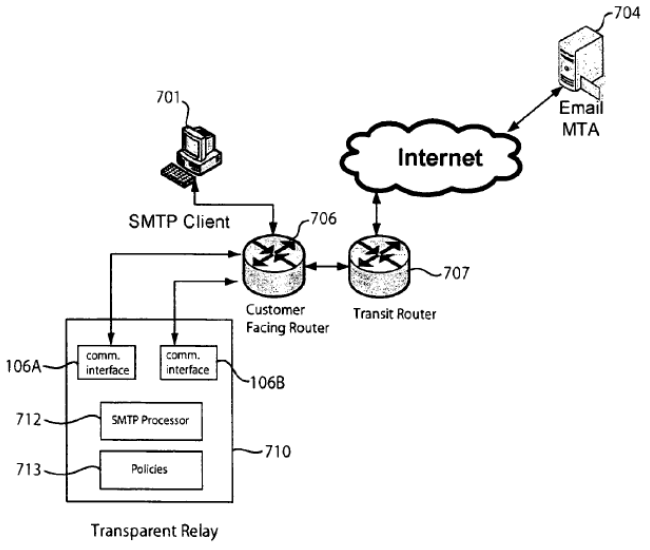


FIG. 9

Figure 3 reproduced from Rand et al.

Rely 706 which hosts the "transparent relay 710" may use SMTP port 25, which is the standard go-to port for email (Rand para. 0086.) The relay 710 receives the email to and from client 701 and 710 operates as a TCP reply rather than an MTA. (Rand para. 0086.) Therefore, it has some characteristics of an MTA. (Rand 0086, 0087) Accordingly this allows use to perform some level of hardening and policy making on our router device. The relay 710 is not a physical relay but is a series of "logical tunnels" like IPIP or GRE tunnels. (Rand para. 0088.) This allows IP packet to continue to have the same source and destination IP address untouched by the relay 710. (Rand para. 0089.) SMTP processor 712 works in conjunction with the policy table 713 to determine what "email actions are permitted in the network." (Rand para. 0091.) Our system admin can set them up. (Rand para. 0091.) Within 710, we have 713 which is the "Policies" which will be discussed (Rand para. 0087.)

Technical Mitigation — Policy Tables of Router as Hardware

As discussed above, the primary mode of information transfer is the compression, the encryption, and embedding of the secret message into the STMP protocol's header as disclosed in Castiglione. Therefore, our primary means of remedies this is to analyze the header of the email as disclosed by Rand et al. in Fig. 12 Item 833. Para. 0114 of Rand et al. instructs users to "analyze[]" the header of the email. Additionally, prior to doing so, it would be appropriate to analyze the "data state" as this would help block cover channels (Rand para. 0113.)

Additionally in Rand et al., there is a discussion on "email traffic by the email server from the MTA which is to perform an analysis of "malformed" packets to "block covert channels." (Rand para. 0122.)

Importantly, Castiglione discusses the use of spam emails as a means to increase payload and to increase obscurity. (Id. p. 505.) Therefore, we need a counter measure. Rand discloses "aggregate email message analysis" in para. 0138. Therefore, Rand offers an excellence counter measure to the teachings of Castiglione. That is, InfoSec would be able to harden the network by interposing a route with packet analysis and aggregate analysis.

In short, the teachings of Rand help to mitigate the threats outlined in Castiglione.

<u>Policy Mitigation of Covert Channels</u>

Rand disclosed the use of a policy table (Fig. 9 Item 713) to be set up by an admin. (Rand para. 0091.) However, our policies do not need to be limited to device policies. Organizations should additionally have people policies as people are the greatest risk to any organization.

Castiglione disclosed the use of the headers and how to malform them. One way to remedy this is to have a policy of the use of unused header bits because it is very common for protocol not to specify values. (Zander p. 46, 47.) For example, ICMP protocol has "unused code field." (Zander p. 47.) The appropriate remedy would be for the organization to mandate a set value of these fields which would in turn frustrate the embedding process. (Zander p. 47.)

Unused bits is not limited to ICMP. Other examples include: header extensions and padding, IP identification and fragment offset, TCP initial seq, checksum fields, modulating the IP time to live field, and etc. (Zander p. 46-49.) As such, SMTP is to be treated no differently. This process is called “traffic normalization” and any bits that are “reserved” or “used’ should be standardized. (Zander p. 51.)

Additionally, organization could monitor individual computer usage based on the time of day. Therefore, if there was any unusual traffic, this may be a flag for a covert channel. (Zander p. 53, 54.)

The organization should set up networks appropriately by having high security networks only talk to other high security networks. (Zander p. 51.) Similarly, low security networks should be hooked up to low security networks. (Zander p. 51.) In the case of Rand which disclosed a router interposed between the STMP client the SMTP server, it would be best not to have a client hooked up to the internet at all and disallow any type of communication of that sort if the data was sensitive enough. Based on this, the only way to “create” a covert channel would be the use of physical media hook up to a network and the respective data to be physically carried out. However, as discussed before, policy makers could disallow the use of USB devices on a network.

Step 7

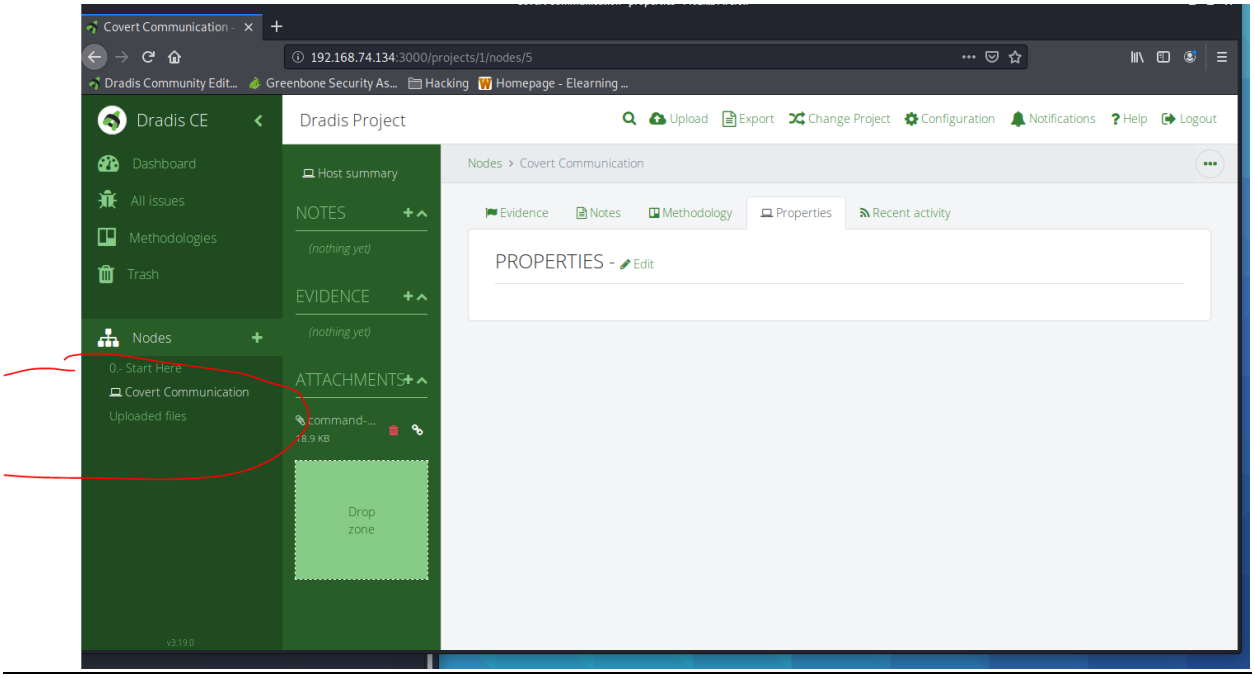


Figure 4 Make Node

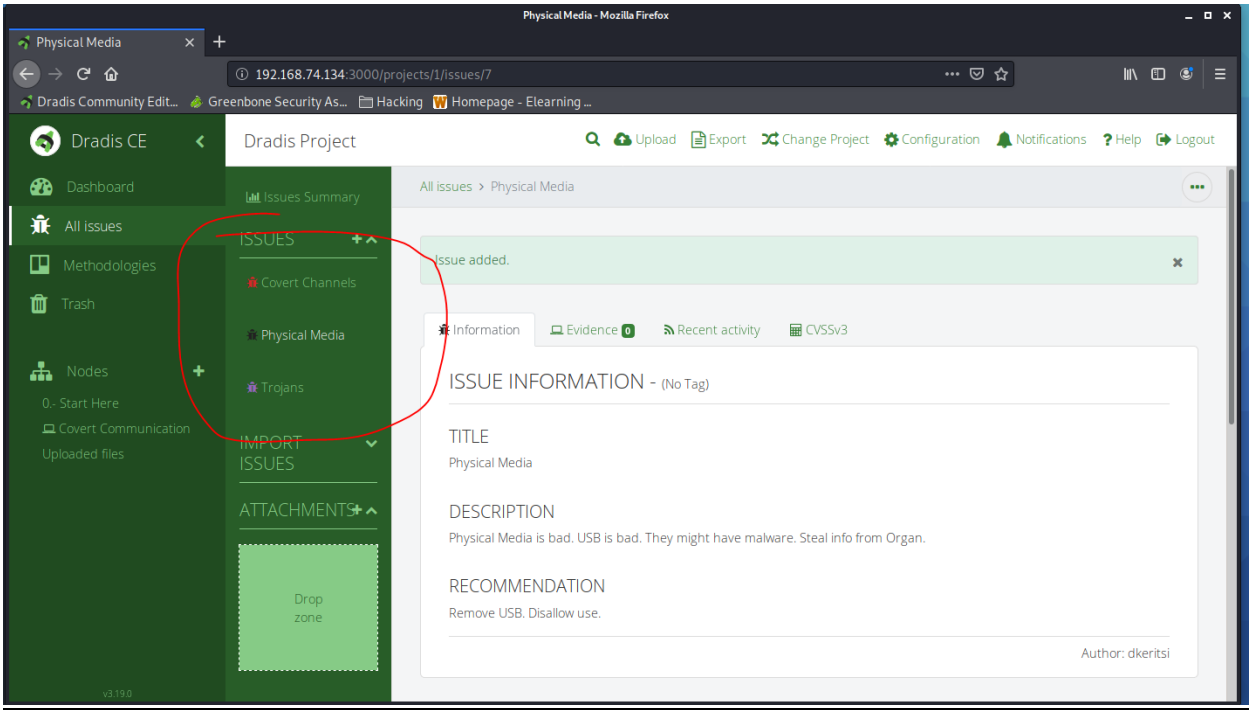


Figure 5 Make 3 Issues and Tag

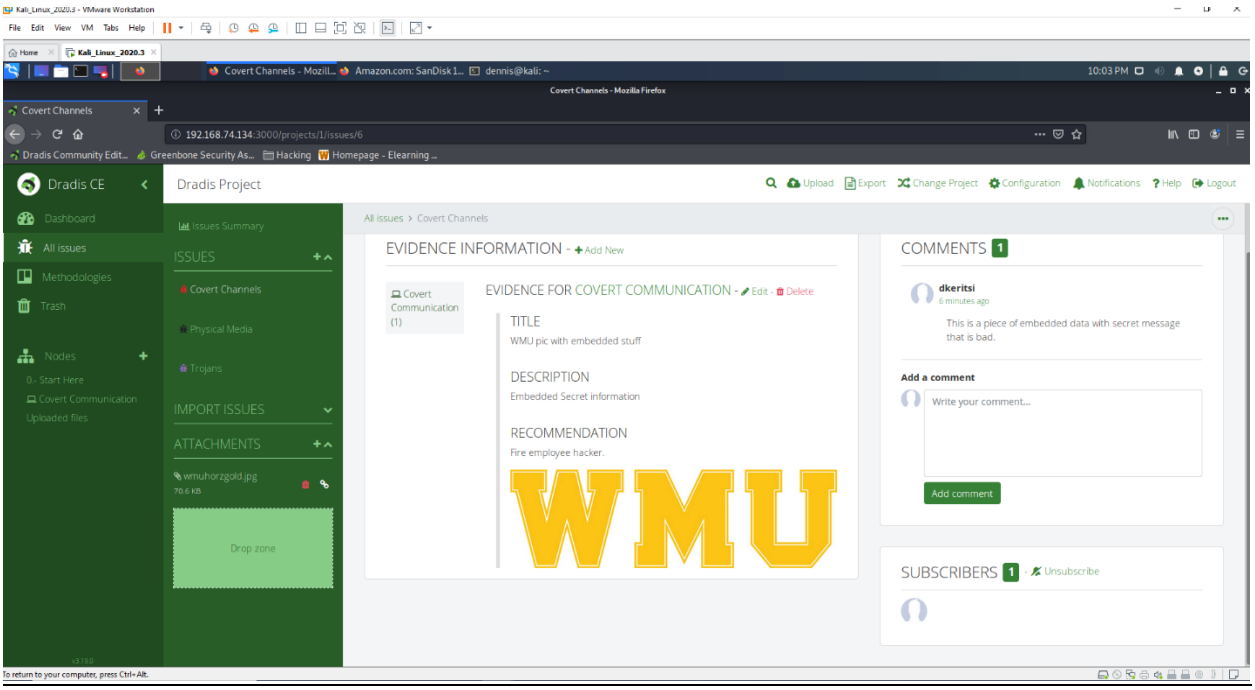


Figure 6 Add Evidence with Pic for First Issue

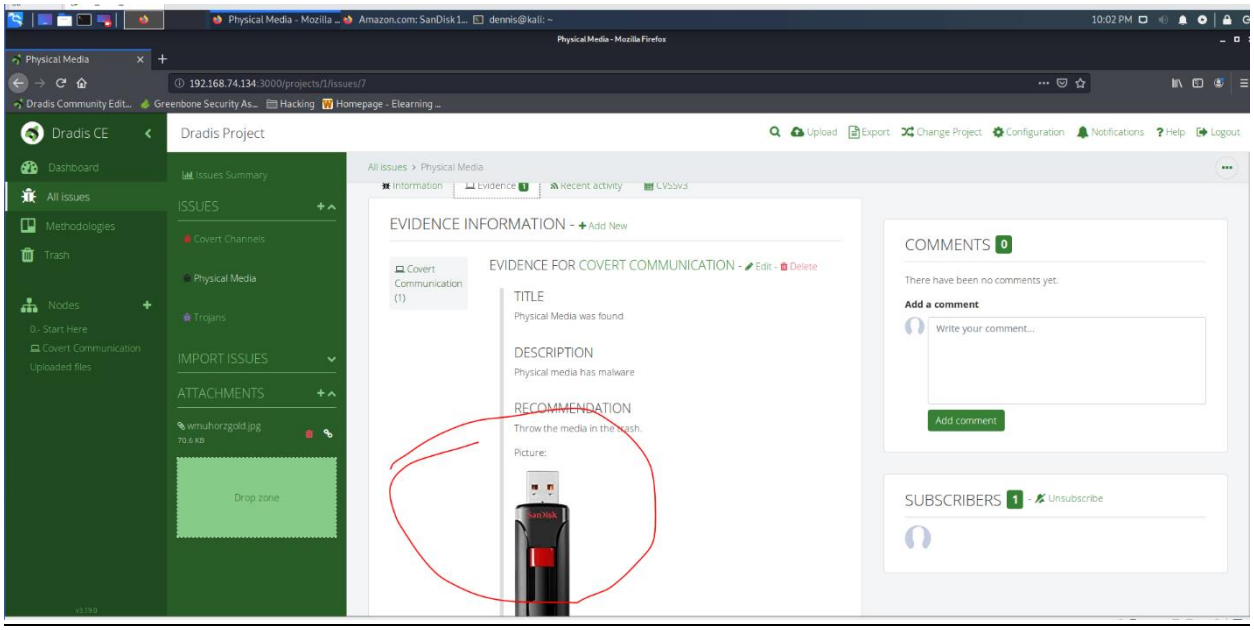


Figure 7 Add Evidence with Pic for Second Issue

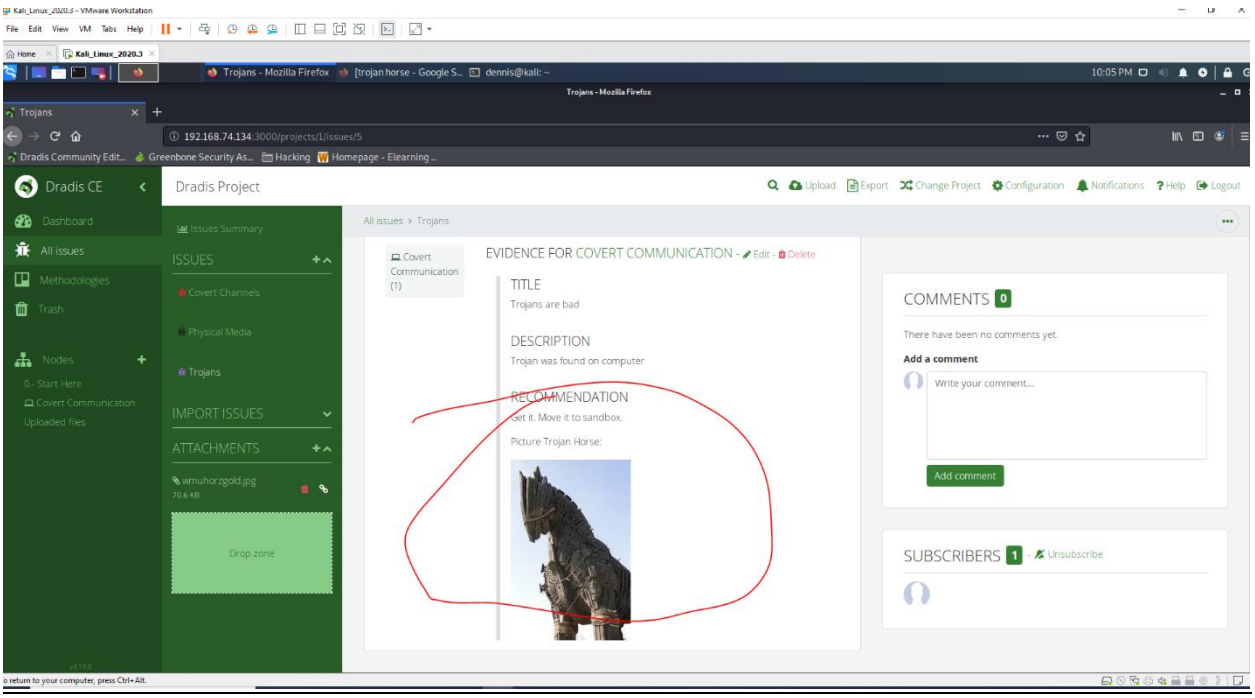


Figure 8 Add Evidence with Pic for Third Issue

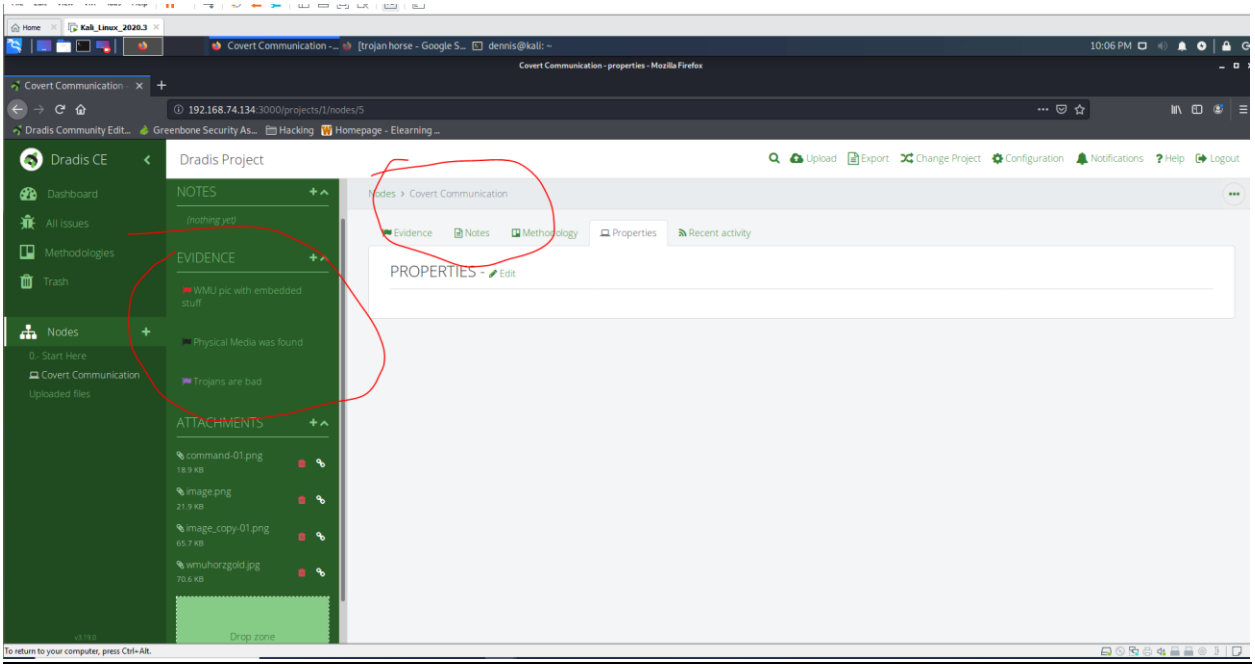


Figure 9 Tag Issues to the Node

References

Castiglione, A., Santis, A. D., Fiore, U., & Palmieri, F. (2011). E-mail-Based Covert Channels for Asynchronous Message Steganography. *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. doi:10.1109/imis.2011.133

Chaerani, W. (2011). INFORMATION LEAKAGE THROUGH SECOND HAND USB FLASH DRIVES WITHIN THE UNITED KINGDOM [Abstract]. *9th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, 5th -7th December 2011*. doi:https://doi.org/10.4225/75/57b2ba7e40cea

Goltz, J. P. (2003, January 23). *Use offense to inform defense. Find flaws before the bad guys do.* [PDF]. SANS Institute.

Heikkila, F. M. (2007). Encryption: Security Considerations for Portable Media Devices. *IEEE Security & Privacy Magazine, 5*(4), 22-27. doi:10.1109/msp.2007.80

Jeong, H., Choi, Y., Jeon, W., Yang, F., Lee, Y., Kim, S., & Won, D. (2007). Vulnerability analysis of secure USB flash drives. *2007 IEEE International Workshop on Memory Technology, Design and Testing*. doi:10.1109/mtdt.2007.4547620

Liao, T. (2018). Design of High-Security USB Flash Drives Based on Chaos ... Retrieved October 3, 2020, from https://www.mdpi.com/2079-9292/7/6/82/pdf

Mell, P., Kent, K. A., & Nusbaum, J. (2005). Guide to malware incident prevention and handling. doi:10.6028/nist.sp.800-83

Rand, D. (Aug. 30, 2007). *U.S. Patent No. US 2007/0204341 Al*. Washington, DC: U.S. Patent and Trademark Office.

Rieschick, G. (May 13, 2014). *U.S. Patent No. US 8,726,376 B2*. Washington, DC: U.S. Patent and Trademark Office.

Saracino, A., Sgandurra, D., Dini, G., & Martinelli, F. (2018). MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention. *IEEE Transactions on Dependable and Secure Computing, 15*(1), 83-97. doi:10.1109/tdsc.2016.2536605

Shukla, J. (Jan. 17, 2008). *U.S. Patent No. US 2008/0016339 A1*. Washington, DC: U.S. Patent and Trademark Office.

Tischer, M., Durumeric, Z., Foster, S., Duan, S., Mori, A., Bursztein, E., & Bailey, M. (2016). Users Really Do Plug in USB Drives They Find. *2016 IEEE Symposium on Security and Privacy (SP)*. doi:10.1109/sp.2016.26

Valentino. (2014, March 18). How to Send Email Using Telnet in Kali Linux. Retrieved October 03, 2020, from https://www.hacking-tutorial.com/tips-and-trick/how-to-send-email-using-telnet-in-kali-linux/

Zander, S., Armitage, G., & Branch, P. (2007). A survey of covert channels and countermeasures in computer network protocols. *IEEE Communications Surveys & Tutorials, 9*(3), 44-57. doi:10.1109/comst.2007.4317620

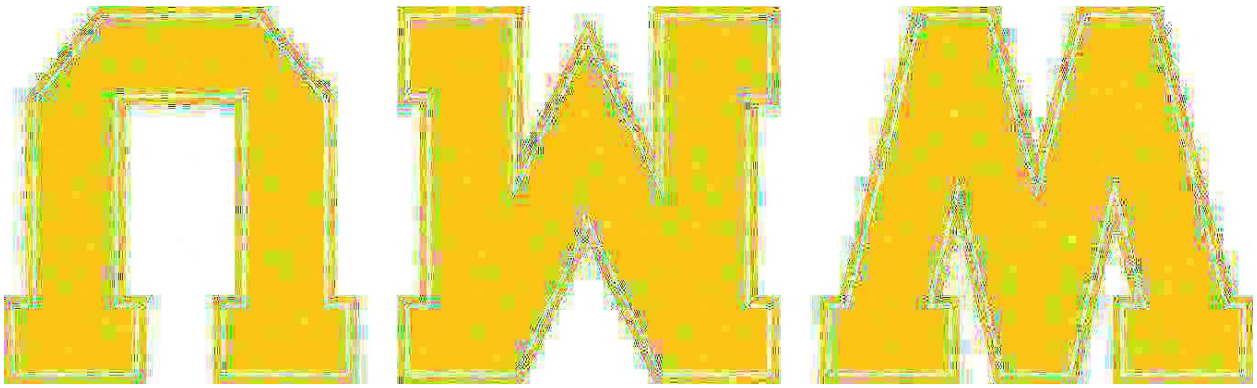
Appendix A

WMU

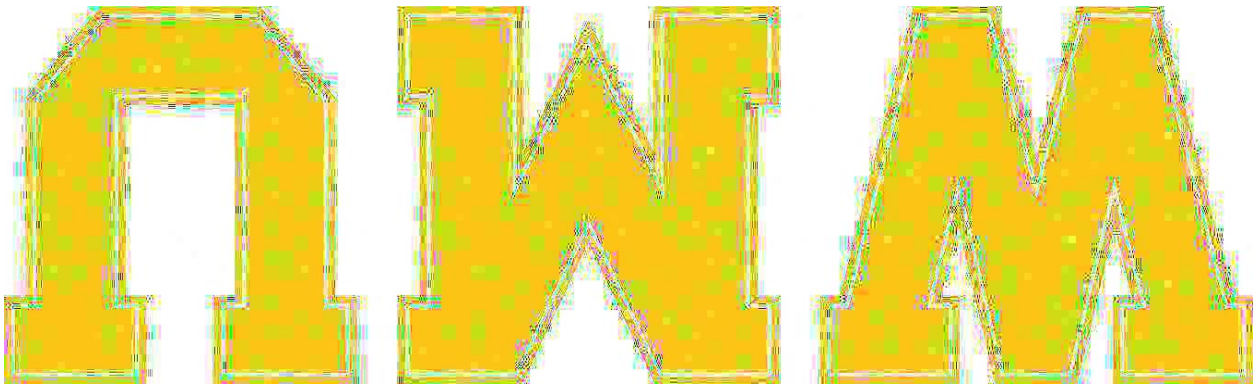
Appendix B


WMU

Appendix C



Appendix D



Appendix E

Appendix F

WMU