1. How many known countries has Regin been identified in?

Ans. Regin has been identified in following 14 countries:

1. Algeria
2. Afghanistan
3. Belgium
4. Brazil
5. Fiji
6. Germany
7. Iran
8. India
9. Indonesia
10. Kiribati
11. Malaysia
12. Pakistan
13. Russia
14. Syria

2. True/False - Regin traffic could be routed from one organization through another organization to reach the Regin command and control servers.

Ans. True

3. Given the following DGA algorithm, what would you do to make it more robust (i.e. make it more difficult to detect)?

```
# create list of TLDs that will be chosen later
tlds = ['com','net','org','ru','cn','tv']
#iterate through the number of domains requested
#create a string that is the date passed in and the interation number
string_to_hash = date + "-" + str(iter)
# create the MD5 hash
hash = hashlib.md5()
hash.update(string_to_hash)
hashed_string = hash.hexdigest()
# determine the length of the domain
# generated based on the numeric value of the first character (mod 10)
length = ord(hashed_string[:1])%10 + 12
# build the final domain
# the first 'length' characters and the TLD determined on the
# first character (mod 6)
return hashed_string[:length]+"."+tlds[ord(hashed_string[:1])%6]
```

Ans 3. We could make following additions in the algorithm to make it more robust:

1. We could use Unix timestamp instead of date
2. We could use pseudo random number generator (PRNG) algorithm to determine the length of domain.
3. We could use PRNG algorithm to build the final domain too instead of using hash.
4. We could use a stronger hash than MD5, such as SHA-512

4. Name three (3) mechanisms, at least one (1) of the three should be subtle, to use non-admin shell access on a workstation in a Microsoft AD network to spread to other systems on the AD network.

Ans 4. Following are the mechanisms to use non admin shell access on a workstation in Microsoft AD network to spread to other systems on AD network:
1. **Pass the hash -** pass the hash is a hacking technique that allows an attacker to authenticate to a remote server/service by using the underlying NTLM and/or LanMan hash of a user's password, instead of requiring the associated plaintext password as is normally the case.
2. **Using stolen ticket** – We can steal a token to gain access.
3. **Vulnerability exploits** – We can scan the network for vulnerabilities and try to gain access with the vulnerabilities found.
4. **Generate Artifact** – We can create executables like SMB beacon to allow us to gain access to the system. Then we wound need to copy the executable to remote host or execute it remotely, which can be done wmic, at, schtasks, sc. Once our code is executed remotely, we would just need to gain control of it.

5. Name three (3) categories of tools that you could use to detect and prevent data exfiltration. How would you implement these tools and what are some of their strengths and weaknesses?

Ans 5. I can think of following Category of tools to prevent data exfiltration:

1. **Network Logging tools** to detect data exfiltration like DNS logs, proxy logs or audit logs. Using DNS and proxy logs we can understand the request and response made by the system to which host or ip addresses. We can analyze the logs to understand if there is any malicious network activity. We can then use this information to block such activity by appropriate action such as blocking a host, ip address or port.
2. **Network Traffic Inspection Tools** to detect data exfiltration through encrypted web sessions, like netflow with splunk. – Similar to logging tools these helps us identify malicious network traffic. Once identified we can block such traffic.
3. **Blocking tools** to block file attachments using signatures or ACLs, firewalls. Using these tools, we can prevent data exfiltration altogether in most of the cases as this restricts access to the system using firewalls and access to the data in the system using file signatures and ACLs.
4. **DB transactions detection tools**: These can help us detect unwanted database transactions which might help us identify the target data.

6. Name and describe three (3) mechanisms an attacker could use to maintain connectivity (command and control) with a compromised host that sits behind a firewall but can access the Internet.

Ans 6. An attacker can use following mechanisms to maintain a C2 channel with the compromised hosts

1. DGA – Attackers can use DGA to send requests to random domains to hide request to C2 server. Once malware sends the request to C2 servers, C2 server can respond to the compromised host to maintain a C2 channel
2. DNS as medium – Malware can target compromise hosts DNS to make requests to C2

server look genuine.
3. P2P – Attackers can use P2P channel to maintain connection to the compromised host.
4. HTTP
5. Protocol Mimicking – Attackers can use protocol mimicking to mimic a protocol which looks genuine in network traffic
6. Using Namecoin service – Attackers can use Namecoin service to create genuine domains to connect to compromise host.
7. Using Esoteric C2 Channels – Attacker can use esoteric channels such as microphone, Bluetooth or any radio at hosts device to connect to compromised hosts.

7. What are the practical differences between static and dynamic malware analysis from a network defender's perspective? How might the results differ?

Ans 7. **Static Malware Analysis:** This is usually done by dissecting the different resources of the binary file and studying each component. The binary file can also be disassembled (reverse engineered) using a disassembler such as IDA. The machine code can be translated into Assembly code which can be read and understood by humans. A malware analyst can then make sense of the Assembly instructions and have an image of what the program is supposed to perform. Analyst can also learn ways to defeat and as a result sanitize the system from the infection of the disassembled malware.
Dynamic Malware Analysis: This is done by watching and logging the behavior of the malware while running on the host. Virtual machines and Sandboxes are extensively used for this type of analysis. The malware is debugged while running using a debugger such as GDB or WinDbg to watch the behavior of the malware step by step while its instructions are being processed by the processor and their live effects on RAM. [Referred from Wikipedia]

Difference between Static and Dynamic Malware Analysis from a Network Defender's perspective:

Static Malware Analysis	**Dynamic Malware Analysis**
Code is not executed, but analyzed.	Code is executed.
Since we don't execute the code, we don't have to worry about creating a safe environment.	Since code is executed, we need to create a safe environment to execute the code to analyze the malware
It is similar to Dissection of dead code.	System is infected to analyze the behavioral changes the system would have to understand the effect malware may have to the system
Static Analysis reveals immediate information on how malware might affect the system	Dynamic Analysis reveals information on what malware is doing rather than how it does it.
Exhaustive static analysis could theoretically answer any questing, but it is slow and hard	It is conducted by observing and manipulating malware as it runs.
We need to look out for the blacklisted strings in the code, we might need to run it through disassemblers to check the way it interacted with memory	During this we can't trust anything that is writable. We need to take snapshots to observe the system. We need to look into Registry activity, File Activity, Process Activity and Network Traffic.

8. What are commands that you could use to obtain domain admin privileges on a domain controller after you use incognito within Metasploit to impersonate a user?

Ans 8. In incognito mode we need to use following commands:

1. **list_tokens –u**: to list available tokens for users that we could impersonate
2. **impersonate token <available token>**: to impersonate the user.

9. True/False - It is possible to run a UDP scan through a SOCKS proxy (like proxychains)

Ans 9. False. It is not possible to tunnel ICMP and UDP traffic via the socks proxy

10. What credential and information is needed to create a Windows authentication Golden Ticket?

Ans 10. A Golden ticket needs:

1. The Domain Name
2. The Domain SID
3. The krbtgt account's nt hash
4. The user account you want to create the ticket for

11. Why should a domain name in DNS bind configuration files end with a '.' (i.e. a single period) character?

Ans 11. Because a fully-qualified (unambiguous) DNS domain names have a dot at the end. A domain name that doesn't have a dot at the end is not fully-qualified and is potentially ambiguous. This was documented in the DNS specification, RFC 1034, way back in 1987:

Since a complete domain name ends with the root label, this leads to a
printed form which ends in a dot. We use this property to distinguish between:

- a character string which represents a complete domain name
  (often called "absolute"). For example, "poneria.ISI.EDU."

- a character string that represents the starting labels of a
  domain name which is incomplete, and should be completed by
  local software using knowledge of the local domain (often
  called "relative"). For example, "poneria" used in the
  ISI.EDU domain.

Referred from http://www.dns-sd.org/trailingdotsindomainnames.html

And in some cases if we have set an $ORIGIN in the bind configuration file, it will be appended at the end if we don't include a trailing period after the domain name in bind configuration file.

12. Can a non-explicitly seeded random number generator be used as the input into a hash function in the generation of a domain for Domain Generation Algorithm (DGA) malware? Why or why not?

Ans 12. Assuming that non explicitly seeded random number generator is an algorithm or a RNG that returns a random number which can be used in DGA instead of using a seeded RNG algorithm's returned number. We can use its output into a hash function in the generation of domain in DGA malware. Because a DGA can return domain even when we don't use seeded RNG algorithm, but in this case we are just replacing a module with another which returns same type of output without any input. So there is no reason DGA wouldn't work with a non-explicitly seeded random number generator.

13. Please describe what the below Python code does.

```
import os
import re
regex = re.compile(r'.*\d{3}\.\d{3}\.\d{4}.*')
for root, dirs, files in os.walk("."):
for file in files:
if file.endswith(".csv"):
print("Found in file: "+os.path.join(root, file))
my_file = open(root+"/"+file,"r")
for line in my_file:
discovered_values = regex.findall(line)
for value_of_interest in discovered_values:
print value_of_interest
```

Ans 13. It is trying to search of patterns in csv file that could be phone numbers or pattern with numbers in xxx.xxx.xxxx format. The regex in the code is to search for any such pattern and then the code is checking each line in csv file and printing those values.