Module 4.1
Zymkey Self Destruct and Binding

CompuThings *Technology;

Module Target

- Understand and perform necessary steps for production binding
- Produce Production Bind

Step-by-step Flow

- Understand and perform necessary steps for production binding
- Run code for perimeter detection
- Configure zymkey to self-destruct
- Cut zymkey cut-2-lock for production bind
- Cut the perimeter detect circuit
- Analyzed the self destruct that occur

Developer Mode Binding

- When the LED blink in the slow mode, zymkey is bind to the Rpi in the development mode.
- In this mode, zymkey is bind but can be move to another Rpi.

Production Mode Binding

- This binding bind Zymkey to specific host Pi and SD Card.
- WARNING: THIS BINDING IS PERMENANT AND CANNOT BE REVERSED. PAY ATTENTION TO THE FOLLOWING
- Your specific Zymkey will be locked to the specific host Pi and it is impossible to move or bind your Zymkey to another Pi. There are no factory resets, masterkeys or other forms of recovery
- If you are using the perimeter_detect features, then the sequence in which you arm, disarm is very important. Be sure to follow the process steps below.

Process for Moving from Developer to Production Mode

1. Install the battery on Zymkey
2. Place Zymkey onto the Pi (with power down on the pi)
3. Turn on the Pi
4. Install and bind the Zymkey and Pi
5. Set Perimeter Event Actions to “none” or “notify only”
6. Create your LUKS encrypted volume
7. Install your applications into your encrypted volume
8. Confirm your system and applications work fully as you intend

Process for Moving from Developer to Production Mode...2

9.Turn off the power to the Pi.

10. Do not remove the battery.

11. Remove the zymkey from the Pi

12. Now Cut the Lock Tab

13. Replace the Zymkey onto the Pi and turn on power to the Pi

14. Close your perimeter circuit(s) (enclosure lid)

15. Clear Perimeter Detect Events

16. Get Perimeter Detect Info to confirm prior events are cleared and the perimeter is closed.

17. If the Perimeter Detect Event returns clear, then you can ‘arm your system’ as you require by setting Set Perimeter Event Actions to “none”, “notify” or “selfdestruct”

18. Your system is now armed.

API for Perimeter Detect

- For perimeter detect events check out:
  - set_perimeter_event_actions()
  - get_perimeter_detect_info()
  - wait_for_perimeter_event()
  - clear_perimeter_detect_info()

Test API using python interactive

- get_perimeter_detect_info()

```
pi@raspberrypi:~ $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import zymkey
>>> zymkey.client.get_perimeter_detect_info()

[1582764209L, 1582764209L]
```

Try with Your Zymkey

- Take Your Cable, connect the wire and Try
- Check the reading. Disable circuit that you didn’t use.

Perimeter Event is detected

- Circuit breach is detected
- Some how our perimeter circuit do not work as we want
- But we still want to ‘see’ how self destruct work.

Set the respond to self-destruct

- This is just for experiment for learning process.
- So we can see how sample of self destruct happening in HSM.

Conclusion

- In the world of pervasive computing, physical security is the first layer of security.
- This function with accelerator sensor is important for physical security.