Alex Sheen
Computer Security
02-03-22
Assignment 3

Problem 1

For this problem, I knew I could find MD5(*api_tag*, '&role=admin), the *api_tag* for my attack URL. The real problem was finding the padding used to obtain the provided *api_tag*. Once I discovered the padding, I could append the padding + '&role=admin' to the URL to obtain a plaintext string that would hash to my attack *api_tag*. I discovered this padding by repeatedly querying FlickUr with paddings until the correct one was accepted by FlickUr. I didn't use any additional ideas or techniques besides a reliance on Python's .encode, .decode, and bytes.fromhex to manipulate the various pieces of data.

Hex of file:

696d706f72742075726c6c69622e726571756573740a696d706f72742062
61736536340a0a66726f6d2070796d643520696d706f7274206d64352c20
70616464696e670a0a232323232323232323232323232323232323232323
232323232323232323232323232323232323232323232323232323232323
232323232323232323232323232323232323232323232323232323230a
23200a232054686973207374617274657220666696c6520666f7220554368
696361676f20434d534320323332303020 2f203333323530206973206 20666f
7220507974686f6e330a230a232323232323232323232323232323232323
232323232323232323232323232323232323232323232323232323232323
232323232323232323232323232323232323232323232323232323232323
23230a0a232323232323232323232323232323232323232323232323232323
232323232323232323232323232323232323232323232323232323232323
232323232323232323232323232323232323232323232323230a23200a2320
6d616b655f717565727928636e65745f69642c207175657279290a23202d
2d20636e65745f69642073686f756c6420616c7761797320626520796f75
72206f776e20636e65745f69640a23202d2d207175657279206361 6e2062
6520616e7920737472696e67206f662062797465732c20696e636c756469
6e67206e6f6e2d7072696e7461626c650a230a23204d7573742062652072
756e20226f6e2063616d70757322206f722077697468696e207468652056
504e2c206f74686572276973652069742077696c6c2068616e672e0a230a
232323232323232323232323232323232323232323232323232323232323
232323232323232323232323232323232323232323232323232323232323
232323232323232323232323232323232323230a0a534552564552203d
2022687474703a2f2f736563757269796 36c6173732e63732e75636869
6361676f2e6564752f220a0a646566206d616b655f717565727928636e65
745f69642c207175657279293a0a2020202044454255472 03d2046616c73
653b2023205265706c61636520776974682022547275652220746f207072

696e74206578747261206465627567676e6720696e666f726d6174696f
6e0a20202020636e65745f6964203d20636e65745f69642e6c6f77657228
290a2020202069662044454255473a200a2020202020202020207072696e74
282251756572796e67207468652073657276657222290a202020202020
20207072696e74282228434e45542049443a222c20636e65745f69642c20
222922290a2020202020202020207072696e74282228512851756572793a222c20
71756572792c20222922290a2020202069662028747970652871756572792
92069732062797465617272617929206f722028747970652871756572792
92069732062797465732933a0a2020202020202020207572 6c203d20534552
564552202b2075726c6c69622e70617273652e71756f74655f706c757328
636e65745f696429202b20222f22202b2075726c6c69622e70617273652e
71756f74655f706c75732826173653 42e75726c736166655f62363465
6e636f646528717565727929292 02b20222f220a20202020656c73653a0a
2020202020202020207572 6c203d2053455256455220 2b2075726c6c69622e
70617273652e71756f74655f706c757328636e65745f696429202b20222f
22202b2075726c6c69622e70617273652e71756f74655f706c7573286261
736536342e75726c736166655f623634656e636f646528717565727 92e65
6e636f646528277574662d3827292929202b20222f220a20202020696620
44454255473a0a2020202020202020207072696e74282228 5175657279696e
673a222c2075726c2c20222922290a2020202077697468 2075726c6c6962
2e726571756573742e75726c6f70656e2875726c29206173207265726f
6e73653a0a20202020202020616e73776572203d206261736536342e7572
6c736166655f6236346465636f646528726573706f6e73652e7265616428
29290a2020202020202072657475726e20616e737765720a0a2323232323
2323232323232323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323232323
232323232323232323232323230a232048656c706572206d6574686f
64732c206966206e65656465642c20676f2062656c6f7720686572650a23
2323232323232323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323232323
6520686572650a2323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323230a23
2050524f424c454d203120534f4c5554494f4e0a23232323232323232323
2323232323232323232323232323232323232323232323232323232323
2323232323232323232323232323232323232323232323232323232323
2323232323232323230a0a6465662070726f626c656d3128293a0a2020
2020636e6574203d2022616c6578736865656e220a0a2020202075726c20
3d206d616b655f717565727928636e65742c202222290a20202020707269
6e7428225c6e53746172742055524c3a2022202b2075726c2e6465636f64
6528275554462d382729290a0a20202020706172616d735f737472203d20
706172616d735f73747273203d2075726c2e6c737472697028622768747
703a2f2f7777772e666c69636b75722e636f6d2f3f27290a202020207073

203d206c697374286d6170286c616d62646120783a20782e73706c697428
62273d27292c20706172616d735f737472732e73706c6974286227262729
29290a20202020706172616d73203d207b705b305d3a20705b315d20666f
72207020696e207073206966206c656e287029203e3d20327d0a20202020
707265765f746167203d20706172616d735b62276170695f746167275d0a
0a202020207072696e7428225c6e707265765f7461673a2022202b207072
65765f7461672e6465636f646528275554462d382729290a0a2020202068
203d206d64352873746174653d62797465732e66726f6d68657828707265
765f7461672e6465636f646528275554462d382729292c20636f756e743d
353132290a20202020682e757064617465286226726f6c653d61646d69
6e27290a202020206e65775f646967657374203d20682e68657864696765
737428290a0a202020206e65775f75726c203d206227687474703a2f2f77
77772e666c69636b75722e636f6d2f3f6170695f7461673d27202b206e65
775f6469676573742e656e636f646528275554462d382729202b20622726
756e616d653d616c6578736865656e26726f6c653d7573657226726f6c65
3d61646d696e270a0a2020202069203d20300a202020207768696c652028
693c3d393939393939293a0a20202020202020207072696e742869290a20
202020202020206e65775f75726c203d206227687474703a2f2f7777772e
666c69636b75722e636f6d2f3f6170695f7461673d27202b206e65775f64
69676573742e656e636f646528275554462d382729202b20622726756e61
6d653d616c6578736865656e26726f6c653d7573657227202b2070616464
696e67286929202b20622726726f6c653d61646d696e270a0a2020202020
202020696620286d616b655f717565727928636e65742c206e65775f7572
6c2920213d206227496e636f72726563742068617368827293a0a20202020
2020202020202020207072696e74282268697742122290a2020202020202020
202020207072696e74286d616b655f717565727928636e65742c206e6577
5f75726c292e6465636f646528275554462d382729290a20202020202020
2020202020726574756e206e65775f75726c0a2020202020202020200a20
202020202020202069202b3d20310a0a202020207072696e74282266616c
656422290a2020202072657475726e206e65775f75726c0a0a2320436f64
652062656c6f772068657265207696c6c2062652072756e20696620796f
752065786563757465202770797468686f6e332061737369676e6d656e7433
2e7079272e0a232054686973697320636f6465206865726520776f6e27742062
65206772616465642c20616e6420796f757220636f64652061626f766520
73686f756c646e277420646570656e64206f6e2069742e0a6966205f5f6e
616d655f5f203d3d20225f5f6d61696e5f5f223a0a202070726f626c656d
3128290a202023206f7074696f6e616c206472697665722064652068
6572650a20206578697428290a

Problem 2

Running time: 1.66655 s
real 0m1.685s

user 0m1.672s
sys 0m0.001s

alexsheen-2a-1.bin

616c6578736865656e2074686973206973206d792066696c650a00000000
0000000000000000000000000000000000000000000000000000000000000000
000000004c2f5d80c0818305ccca810746fe09ecddced51620df9fecff0d
39b6eb1b98ca8e0c00cb5802a8476b04655b02b662d08c1a9d11db20365c
2e0028808f67f574eb63794259bf9d4b1a89cee7ac3a736f4163e149efac
282f7cb79f4b5b1d65c1eafebc2d84288cd66cc06f518b3b6ff18eed87d1
2e6ad4deeb6b07e371ba69b2

alexsheen-2a-2.bin

616c6578736865656e2074686973206973206d792066696c650a00000000
0000000000000000000000000000000000000000000000000000000000000000
000000004c2f5d80c0818305ccca810746fe09ecddced59620df9fecff0d
39b6eb1b98ca8e0c00cb5802a8476b04655b023663d08c1a9d11db20365c
2e0028008f67f574eb63794259bf9d4b1a89cee7ac3a736f4163e1c9efac
282f7cb79f4b5b1d65c1eafebc2d84288cd66cc06f518bbb6ef18eed87d1
2e6ad4deeb6b076371ba69b2

MD5(alexsheen-2a-1.bin)= a5a797289a4c5eb6bd191a030f101a33
MD5(alexsheen-2a-2.bin)= a5a797289a4c5eb6bd191a030f101a33

SHA256(alexsheen-2a-1.bin)=
726acc7b895a02887a6c7dfcd93519383512149ade69b00d65602ddd5fe0ff6e
SHA256(alexsheen-2a-2.bin)=
1a31ac3bca3aae6d7449d70d455b7b0d32eaa763da8b14db83222f1620bafd60

Problem 3

https://stackoverflow.com/questions/4749330/how-to-test-if-string-exists-in-file-with-bash
https://stackoverflow.com/questions/3953645/ternary-operator-in-bash

MD5(alexsheen-2b-1.sh)= dd253009a6b1f39678f6e10ceefa1afe
MD5(alexsheen-2b-2.sh)= 720a9c2c0f0905dc14844fb0712f2b1d

Run either file:
./alexsheen-2b-1.sh
./alexsheen-2b-2.sh

To create this exploit, I started with a file text.sh which would represent my prefix. It contained the first line of the here-document code.

I then ran fastcoll on text.sh to produce alexsheen-2b-1.sh and alexsheen-2b-2.sh, two files with the same MD5 hash. This represented my prefix + blob1 and prefix + blob2.

I then appended ‘EOF’ to both files.

The lines cat << ‘EOF’ > outfile and ‘EOF’ essentially sandwich and contain the binary blobs created by fastcoll. They are safely stored in junk outfiles.

I also appended code to print “I am good/evil” based on a sequence of bytes that appear on one blob but not the other. In this case, I was able to find the string “YN”, which I can search for using ‘grep’. I then switch based on ‘grep’ to decide whether to send good/evil.