

season of KUPC

고대의 봄

<div>0</div> <div>20DDBC0</div> <div>난독이 귀찮아</div>	<div>F0FF000100002A04</div> <div>IED8043000 → F0FF000100002C04</div> <div>IED80433008 → 00000000 인듯 0 이니 그대로 (or 20DDC32)</div>
<div>1</div> <div>20A22B4</div> <div>변조 방지 값 생성</div>	<div>D3D12SDKPath + F638 → 21A5537568B3</div> <div>rsp B821CFF330 과 xor 해서 211D72BA9B83</div> <div>8000000200000000 상수값</div> <div>IEC3B2513E0 → IED79AC0000 0 아니니 그대로 (or 20A2380)</div>
<div>1</div> <div>20A22FB</div>	<div>IEC3B2513F8 → 800</div> <div>IED8043000 → F0FF000100002C04) 난독이해서 2A8 2F9</div> <div>IEC3B2513E0 → IED79AC0000 + 2A8 * 4 → 3E3 40F</div> <div>즉 IED79AC0AA0 음수 아니니 그대로 (or 20A2380)</div>
<div>1</div> <div>20A2340</div> <div>원본이 2바이트</div>	<div>IEC3B2513E8 → IED79AC2000 + 3E3 * 2 * 8 + 08 → F0FF000100002C04</div> <div>즉 IED79AC5E38</div> <div>같은 20A235A (or 그대로)</div>
<div>1</div> <div>20A235A</div>	<div>IEC3B2513F0 → IED79ACA000 + 3E3 * 8</div> <div>즉 IED79ACBF18</div> <div>즉 값이 0이 아니니 그대로 (or 20A2380)</div>
<div>2</div> <div>52F7C2</div>	<div>상수 8000000200000000</div> <div>x2 srl vs +1</div> <div>크면 52F885 (or 그대로)</div>
<div>2</div> <div>52F885</div> <div>난독이후에 귀찮았기</div>	<div>IED33EF6310</div> <div>IEC3B2513F0 → IED79ACA000 + 3E3 * 8 → IED33EF5D10</div> <div>8000000200000000 넣기 +1</div> <div>B821CFF358 → B821CFF368 + 0 * 8 → IED33EF5D10 저장</div> <div>F0FF000100002C04 vs FF</div> <div>같은 20A244D (or 20A244D)</div>

season of 2021

고대의 봄

<p>' 20A238B</p>	<p>FOFF000100002C04 \downarrow or FO \downarrow 상수 상수 FO FF FFFF 00002C04 저장 IEC3B2513E0 \rightarrow IED79AC0000 0 아니니 그대로 (or 20A244D)</p>
<p>' 20A23B5</p>	<p>IEC3B2513F8 \rightarrow 800) 난독화해서 95 2C4 FOFFFFFF00002C04 IEC3B2513E0 \rightarrow IED79AC0000 + 95 * 4 \rightarrow FFFFFFFF 음수니 20A244D (or 그대로)</p>
<p>위와 같은 예외 약면</p> <p>' 20A244D</p>	<p>8000000200000001 1번째 0이니 20A247C (or 그대로)</p>
<p>' 20A247C</p>	<p>B821CFF3E8 \rightarrow 0000000000000000 상수 저장 8000000200000001 음수니 20A249E (or 그대로)</p>
<p>' 20A249E</p>	<p>B821CFF378 \rightarrow 211D72BA9B83 난독화해서 넣었을 거 rsp B821CFF330 과 같이 xor 해서 21A5537568B3 원래 값과 동일하니 그대로 (or 2425B59)</p>
<p>' 2425B49</p>	<p>000021A5537568B3 0이니 그대로 (or 2425B55)</p>
<p>' 20A21CD</p>	<p>B821CFF3E0 \rightarrow IED33EF5D10 0 아니니 20A21DD (or 그대로) 0 아니니 그대로 (or 20DDCA5)</p>
<p>' 20BE810</p> <p>진짜 키 값과 해동러 선택</p>	<p>IEDACFC0D18 EC46D2BF1C IEDACFC4D18 \rightarrow EC46D2BF1C \rightarrow ? IE jmp Overwatch.exe + 3E39960 + IE * 8</p>

season of K2

고대의 봄

² 204F040

IEDACFC4D18 → IEC46D2BF1C → ? 0000091E

2자리 shift 00000009
(8bit)

00000009 4자리 shift 관계 값이 있다면 $nl=1$ 20B6130
없으면 $nl=0$ 204F0

² 204F079

7B5E10A9E01 잘 읽으니 그대로 (or 204F0B0)

² 204F07E

0009 vs FFFF 다르니 그대로 (or 204F0B0)

³ 20AC627

00000009 -1 vs 7A 작거나 같으니 그대로 (or 20AC654)

³ 20AC62E

키 값대만 두번재
핸들러 선택

Overmatch.exe + 8 + 020AC67C → 00 참고 00, 01, CC 만 있음

Overmatch.exe + 00 * 4 + 020AC674 → 20AC64D

jmp Overmatch.exe + 20AC64D

³ 20AC64D

문제가 없으면 그대로 (or 204F0B0)

³ 20A9DB0

IEDB0433018 → 7B5E10A9E01 다시 2바이트 00 이니 그대로 (or 20A9E25)
00000001 다시 2바이트 01 이니 20A9E13 (or 그대로)

³ 20A9E13

값 주스 뽑기

최상위 바이트 유자해서 반바이트 shift

7B5E10A9E01 ⇒ IED7842A780

B821CFF340 → 0000000000000000 초기화

00000009 -1 vs 7A 작거나 같으니 그대로 (or 20AC2F7)

⁴ 20AC105

키 값대만 세번째
핸들러 선택

Overmatch.exe + 8 + 020AC33b → 06 참고 00~0B, CC 만 있음

Overmatch.exe + 06 * 4 + 020AC300 → 20AC1F9

jmp Overmatch.exe + 20AC1F9

이항한 예제
방식

예제 방식

season of K2

고대의 봄

4 20AC1F9

실제로 값 저장

B821CFF308 → IEDACFC4D40 저장

B821CFF300 → IED7842A780 → xmm0 ~ IEDACFC4D40 이 저장
90 50

B821CFF2F0 → B821CFF348 → B821CFF340 → IEDACFC4D40 저장

문제 없으면 20AC8C4 (or 20AC7C)

3 20AC8C4

문제 없으면 20AC7C (or 20AF0C4)

2 204F0B7

새로운 값 주소

난독화해서 내기

IEDACFC0D40

IEDACFC0D30

IEDACFC4D40 *4 +1 한걸 IEDACFC4D30 에 저장

즉 7B6B3F13501 7B6B3F03501

IEDACFC4D20 → IEDACFC4C40 0 아닌 20AC7C (or 20B5F37)

1 20B5F2B

두번째 값 저장 원형

IEDACFC4D28 → B1

IEDACFC4D20 → IEDACFC4C40 → IE<46D2BE88 → ? 14

jmp overwatch.exe + 3E33790 + 14 * 8

season of 21

고대의 봄

<div> <div>2062890</div> <div>20628C7</div> <div>206291E</div> <div>20629CF</div> </div>	BI 87는 shift 한 0000
	IEDACFC4CE8 → IEDACFC4CF8 + 0000 * 2 → ? 2 00 은 20628B0
	IEDACFC4CB0 + ? ≤ 0 이니 206291E
	BI < 0154 이니 206291E (or 2062A61)
	Overmatch.exe + 2F38ED0 + BI * 3 * 8 0 이니 20629CF (or 2062A61) 즉 2F3A7B4
<div>20629E6</div> <div>시계거꾸로</div>	Overmatch.exe + 2F38ED0 + BI * 3 * 8 + 4 → 00000009 27비트 값만 살리고 9자리 shift 한게 값이 있으면 2062 (16비트) 없으면 2062A35
2062A35	IEDACFC4C08 → ≠ FFFFFFFF 424 2062A61 (or 2062A61)
20B2682	Overmatch.exe + 2F38ED0 + BI * 3 * 8 → 00B1 왼쪽으로 8자리 shift BI 00000000 (32bit)) or 2 BI 02E20004 IEDACFC4CD8 → 02E20004
	B821CFF538 → ? ≠ 0 이니 84550D (or 2062A61)
84550D	B821CFF330 → 00 저장 BI 02E20004 반독니해서 6A3D26DE) and 해서 2 B821CFF550 → 00000004 - 01 B821CFF538 → IEDB04377E0 + 2 * 4 → FFFFFFFF 음수니 845584
845584	B821CFF55C → 00000006 424 이거나 274 이니 8455B5

season of K2

고대의 봄

³ 8455B5

B821CFF554 →
B821CFF550 → 00000004) 다르니 8455D9

³ 8455D9

B821CFF554 → 00000001 1바이트 치환하기

문자열은 4바이트 shift 00000010
(4바이트)

B821CFF548 → 1EDB0437830 + 1 * 4 → 00000000 저장

B821CFF540 → + 10 → 643026DE 저장

" + 4 → FFFFFFFF 저장 (1EDB04377E8에 있는거)

" + 8 → B102E20004 저장

1EDB04377E8 → 00000001 저장

00000009 의 1B 바이트 바이트 값 0014 20B26CB
(29)

² 20B26CB

B821CFF330 → 00 vs 00 같은지 체크 (or 20B271A)

² 20B26D6

B821CFF528 → 1EC3B2542F0 + 8 → 00000000 + 1

" + C → 0006013C x2 sarl

or 014 20B2043

1EDB0437834 → 저장

² 20B2043

00000000 + 00000000 * 4 = 0

1EC3B2542F0 → 1ED743A4010 + 0 * 8 + 00 → 0000000000000000 2 바이트 1ED4CF4CA0

" + 08 → 00000000 FFFFFFFF 2 바이트 0002E20004

+ 10 → 00000000 00000000 2 바이트 00B10002 FFFFFFFF8

+ 18 → " 7B48C6C0041

+ 20 → " 7B5E10A9E01

Overmatch.exe + 2F38ED0 + B1 * 3 * 8 → 00000009 000200B1

1EC3B2542F8 → 00000000 1바이트 치환하기

season of K2

고대의 봄

' 20A24C0	$B82CFF3E0 \rightarrow 1ED33CF6298 + 10 \rightarrow 0$ $1ED33EF5D10$ ← $20A24D3$ 옳음 (or 그대름) ↕ $20A24D3$
' 20A24D3	$B82CFF3E0 + 08 \rightarrow 0$ 옳음 그대름 (or 20DDBF0 옳음)
' 20DDCA5	$1EDB0433000$ vs $1EDB0433040$ 다르니 20DDBC0 옳음 (or 그대름) $0400 + 20$